Appendix A

Solaris 2.4 Security Checklist

Topic: AUDIT
SubTopic: Audit of Unsuccessful login attempts

Objective 273

Verify that unsuccessful login attempts are logged.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:
Verify the following file exists:

```
/var/adm/loginlog
```

Expected Results:
The `/var/adm/loginlog` file should exist on the host.

Comments:
Unsuccessful attempts to log into the system can be recorded. If the /var/adm/loginlog file does not exist, nothing is logged.

Step: 2

Required Action:
Attempt to login using an INVALID password on a VALID user account. Repeat this step 4 times. Browse the /var/adm/loginlog file.

Expected Results:
An entry exists in the file detailing the unsuccessful login attempts.

Comments:
After a user makes five consecutive unsuccessful attempts to log in, all attempts are recorded in the file /var/adm/loginlog. If a user makes fewer than five unsuccessful login attempts, none of them are logged. If the /var/adm/loginlog file does not exist, nothing is logged.

Step: 3

Required Action:
Attempt to login using an INVALID password on an INVALID user account. Repeat this step 4 times. Browse the /var/adm/loginlog file.

Expected Results:
An entry exists that reports the unsuccessful login attempts.

Comments:

Topic: AUDIT
SubTopic: Defined Audit Events

Objective 272

Verify that the kernel audit events have not been modified inappropriately.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Browse the following file:

```
/etc/security/audit_event
```

Compare its contents with the default kernel audit events file supplied with Solaris 2.4.

Expected Results:

Any kernel audit event modifications must be justified.

Comments:

The system actions that are auditable are defined as audit events in the /etc/security/audit_event file. Each auditable event is defined in the audit_event file by a symbolic name, an event number, a set of preselection classes, and a short description.

Topic: **AUDIT**
SubTopic:

Objective 14

Ensure the audit subsystem is enabled.

Rationale:

UNIX maintains a number of log files that keep track of when users log in and which commands they run. These log files form the basis of UNIX's auditing system. Auditing can be enabled or disabled. It should always be enabled for a secure system.

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

Type in the following command to verify if auditing is enabled:

```
#auditconfig -getcond
```

Expected Results:

The `-getcond` option obtains the machine auditing condition. The response is one of three possible conditions:

auditing - Auditing is enabled and turned on
no audit - Auditing is enabled but turned off
disabled - The audit module is not enabled

An error message with the format "auditconfig: error = Invalid argument(22)" indicates that the BSM option has not been enabled on the system and the auditconfig command cannot be used.

Comments:

Basic Security Module should be installed and turned on.
The -chkconf option of the auditconfig command checks the configuration of kernel audit events to class mappings and reports any inconsistencies.

Topic: **AUDIT**
SubTopic:

Objective 196
Verify the system is capable of detecting when the audit file reaches a configurable threshold and audit records are not lost if this threshold is reached. If the audit file becomes full, verify the system is shutdown until the audit data is archived.

Rationale:
DII COE SRS Requirement:
3.2.2.1.3 The COE shall be capable of detecting when the audit trail reaches a configurable threshold (i.e., % of capacity), ensuring that audit data is not lost, and maintaining system availability.

Test Actions:
Step: 1

Required Action:
Type in the following command:

```
      #auditconfig -getpolicy
```

Fill up the partition holding the audit data (location of audit can be found in the /etc/security/audit_control file). Add space to bring total usage to 100%. Use the 'mkfile' command to generate space. This should result in mail being sent to the isso (more accurately the audit_warn mail alias on the local host which should point to the isso's normal email address).

Expected Results:
The command should return

```
    audit policies = ?
```

where the ? does not include "cnt".

Email in the system administrator’s normal mail folder indicating an audit error had occurred on the machine.

The cnt policy flag should not be enabled ensuring that processes will suspend when audit resources are exhausted.

Comments:
The auditconfig command provides a command line interface to get and set kernel audit parameters. The -getpolicy parameter causes the kernel audit policy to be displayed. If the cnt policy flag is enabled, the kernel is directed not to suspend processes when audit resources are exhausted. Instead, audit records are dropped and a count is kept of the number of records dropped. By default, processes are suspended until audit resources become available.

Step: 2

Required Action:
The threshold for the warning message is set in the file /etc/security/audit_control in the 'minfree' line. Adjust this value appropriate to site requirements.

Expected Results:
A properly tuned audit partition that will send email to the system administrator when the audit

partition begins to fill up.

Comments:

Topic: AUDIT
SubTopic:

Objective 17

Verify the system provides the SGSO with a capability to select and enable auditable events including use of I&A, introduction of objects into a user's address space, deletion of objects, trusted user actions, print use, etc.

Rationale:

DII COE SRS Requirement:

3.2.2.2 The COE shall provide the SGSO with a capability to select and enable auditable events.

3.2.2.3 The COE shall be able to audit the following types of events:

3.2.2.3.1 Use of I&A mechanisms
3.2.2.3.2 Introduction of objects into a user's address space (e.g., file open, program initiation)
3.2.2.3.3 Deletion of objects
3.2.2.3.4 Actions taken by trusted users
3.2.2.3.5 Production of printed output
3.2.2.3.6 Other security relevant events.

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
   vi /etc/security/audit_class
```

Compare its content with the default kernel audit events file supplied with Solaris 2.4.

Expected Results:

Files match indicating that audit classes have not been modified inappropriately. Any audit class modifications must be justified.

```
#
# User Level Class Masks
#
# Developers: If you change this file you must also edit audit.h.
#
# File Format:
#
#    mask:name:description
#
0x00000000:no:invalid class
0x00000001:fr:file read
0x00000002:fw:file write
0x00000004:fa:file attribute access
0x00000008:fm:file attribute modify
0x00000010:fc:file create
0x00000020:fd:file delete
0x00000040:cl:file close
0x00000080:pc:process
0x00000100:nt:network
```

```
0x00000200:ip:ipc
```

Topic: **AUDIT**
SubTopic:

```
0x00000400:na:non-attribute
0x00000800:ad:administrative
0x00001000:lo:login or logout
0x00004000:ap:application
0x20000000:io:ioctl
0x40000000:ex:exec
0x80000000:ot:other
0xffffffffffff:all:all classes
```

Comments:
Each audit event is defined as belonging to an audit class or classes. By assigning events into classes, an administrator can more easily deal with large numbers of events. When naming a class, one simultaneously addresses all of the events in that class. Whether or not an auditable event is recorded in the audit trail depends on whether the administrator preselects a class for auditing that includes the specific event.

Topic: **AUDIT**
SubTopic:

Objective 18
Identify any users for whom auditing has been disabled.

Rationale:
An audit flag is on for all existing users at initial conversion to a trusted system. Auditing for individual users can be disabled.

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Type in the following command:

```
   vi /etc/security/audit_user
```

Identify any user contained in the /etc/passwd file that is not also contained in the /etc/security/audit_user file.

Expected Results:
The file /etc/security/audit_user has a line, beginning with the user's login name, for each authorized user.
No audit class in the audit_control file is listed after a second colon on any user line in the audit_user file.

Comments:
The system audit level applies to all users, unless the user has an entry in the /etc/security/audit_users file. The user audit level overrides the system audit level. The fields in /etc/security/audit_users file are separated by colons and are defined as follows:

```
Field1:       Field2:                Field3:
Username:always audit flags:never audit flags
```

All users should be subject to auditing. A unique identity must be associated with all auditable actions.

Topic: **AUDIT**
SubTopic:

Objective 19

Verify required parameters are identified for each recorded audit event including date and time of event, userid, type of event, success or failure of event, for I&A events, the origin of the request, etc.

Rationale:

DII COE SRS Requirement:

3.2.2.4 For each recorded event, at a minimum the audit record shall identify:

3.2.2.4.1 Date and time of the event
3.2.2.4.2 UserID
3.2.2.4.3 Type of event
3.2.2.4.4 Success or failure of the event
3.2.2.4.5 For I&A events, the origin of the request (e.g., terminal ID)
3.2.2.4.6 For events that introduce an object into a user's address space, and for object deletion events, the name of the object, and in MLS systems, the object's security level.

Test Actions:

Step: 1

Required Action:

As an unprivileged user attempt to view the /etc/security/audit_control file using the command:

```
#/usr/bin/more /etc/security/audit_control
```

Expected Results:

Permission to view the file is denied to an unprivileged user.

Comments:

The audit_control file lists audit file systems and audit configurations for the audit daemon: auditd. Each line consists of a title and a string, separated by a colon. The system administrator defines four kinds of lines in the audit_control file:

- The audit flags line (flags:) contains the audit flags that define what classes of events are audited for all users on the machine.

- The non-attributable flags line (naflags:) contains the audit flags that define what classes of events are audited when an action cannot be attributed to a specific user.

- The audit threshold line (minfree:) defines the minimum free space level for all audit filesystems. The minfree percentage must be greater than or equal to 0. The default is 20%.

- The directory definition lines (dir:) defines which audit filesystems and directories the machine will use to store its audit trail files. (SunSHIELD Basic Security Module Guide)

The audit_user file stores per-user auditing preselection data. Each entry in the audit_user file has the form:

```
username:always-audit-flags:never-audit-flags
```

(Solaris 2.4 audit_user man page)

Topic: **AUDIT**
SubTopic:

Step: 2

Required Action:
As root attempt to view the /etc/security/audit_control file using the command:

```
#/usr/bin/more /etc/security/audit_control
```

Expected Results:
The audit event configuration for accounts on the system shows that at minimum the following events are audited.

(ad) Normal administrative operation
(lo) Login, logout,
(fc) Object creation
(fd) Object deletion
(-fw) Failure to write to a file

Comments:
The audit_control file lists audit file systems and audit configurations for the audit daemon, auditd. Each line consists of a title and a string, separated by a colon. The system administrator defines four kinds of lines in the audit_control file:

- The audit flags line (flags:) contains the audit flags that define what classes of events are audited for all users on the machine.

- The non-attributable flags line (naflags:) contains the audit flags that define what classes of events are audited when an action cannot be attributed to a specific user.

- The audit threshold line (minfree:) defines the minimum free space level for all audit filesystems. The minfree percentage must be greater than or equal to 0. The default is 20%.

- The directory definition lines (dir:) defines which audit filesystems and directories the machine will use to store its audit trail files. (SunSHIELD Basic Security Module Guide)

The audit_user file stores per-user auditing preselection data. Each entry in the audit_user file has the form:

username:always-audit-flags:never-audit-flags

(Solaris 2.4 audit_user man page)

Topic: **AUDIT**
SubTopic: **Protection of Audit Data**

Objective 25
Verify the audit data is protected by the system so that access to it is limited to only those authorized to view the audit data.

Rationale:

DII COE SRS Requirement:
3.2.2.1.1 The audit data shall be protected by the system so that access to it is limited to those who are authorized to view audit data.

Test Actions:
Step: 1

Required Action:
As root, determine the name of the audit files listed in a line starting with "dir:" in the /etc/security/audit_control file. For each "filename", as a NON-privileged user, type the following commands:

```
    ls -l "filename"
    more "filename"
```

Expected Results:
Each command should cause an error message to be returned. Every audit filesystem listed in a line starting with "dir:" in the /etc/security/audit_control file should be accessible only to security administrators.

Comments:

Topic: **AUDIT**
SubTopic:

Objective 20
Verify the system provides the capability to correlate all system, administrative, and audit logs.

Rationale:

DII COE SRS Requirement:
3.2.2.7 The COE shall provide the capability to correlate all system administrative and audit logs (e.g., database management system logs, operating system audit logs, and other system logs).

Test Actions:
Step: 1

Required Action:
View all logs including, but not limited to:

utmp, loginlog, lastlog, sulog, aculog, xferlog, syslog, and the c2 audit logs.

Ensure the date and time is recorded for correlation of audit data between the various audit logs.

Expected Results:
The date and time is included in all audit logs for each audit event recorded.

Comments:

Topic: AUDIT
SubTopic: Audit Reduction

Objective 24
Determine if an audit reduction capability exists. This capability can be either OS provided or an add-on product.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Use the audit reduce feature in Solaris to assist with review of audit.

To review an entire audit file type "`cd /home/audit`" then "`auditreduce *| praudit`". If desired this audit can be redirected into a file by adding "`> filename`" at the end of the command.

Expected Results:
All of the audit records present on the system are displayed on the screen.

Comments:
Audit is sometimes stored in many locations. The location of audit data can be determined by viewing the /etc/audit_control file. The directory location follows the key word "dir:".

This audit can be redirected into a file by adding "`> filename`" at the end of the command. To review the audit records pertaining to a specific user and date, type:

```
   auditreduce -d yyyymmddhhmmss -u userid * | praudit
```

Step: 2

Required Action:
Display the audit for a specific user for a specific date by typing:

```
 "auditreduce -d yyyymmddhhmmss -u userid * | praudit"
```

Note: it is possible to specify the review of all audits before a specific date using the -b option or all dates after a specific date using the -a.

Expected Results:
All the audit records for the date/time and user selected will be displayed to the screen.

Comments:
To review all audit data before or after a specific date, use the -b option or -a option, respectively. To display the audit data for a specific event, use the -c with the audit reduce command. For example, to display all logins that have been audited, type:

```
   auditreduce -c lo * | praudit
```

Topic: AUDIT
SubTopic: Audit Reduction

Step: 3

Required Action:
Display the audit for a specific event by using the -c with the audit reduce command (e.g., display all logins that have been audited with the command:

```
auditreduce -c lo * | praudit
```

Note these audit class identifiers are described in /etc/security/audit_control.

Expected Results:
All the audit records related to logins will be displayed to the screen.

Comments:

The audit class identifiers are described in /etc/security/audit_event.

Topic: **AUDIT**
SubTopic:

Objective 21

Verify the audit_warn script has not been modified inappropriately.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
vi /etc/security/audit_warn
```

Expected Results:

The file: /etc/security/audit_warn has not been changed from the default /etc/security/audit_warn delivered with Solaris 2.4.

Comments:

Whenever the audit daemon encounters an unusual condition while writing audit records, it invokes the /etc/security/audit_warn script. This script can be customized by individual sites to warn of conditions that might require manual intervention, or to handle them automatically. For all error conditions, audit_warn writes a message to the console and sends a message to the audit_warn alias.

Topic: AUDIT
SubTopic:

Objective 22

Verify the audit_warn alias has been configured correctly.

Rationale:

Whenever the audit daemon encounters an unusual condition while writing audit records, it invokes the /etc/security/audit_warn script. This script can be customized by individual sites to warn of conditions that might require manual intervention, or to handle them automatically. For all error conditions audit_warn writes a message to the console and sends a message to the audit_warn alias.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
vi /etc/aliases
```

Expected Results:

An entry should appear for the audit_warn alias, and the alias should be the name of an actual account.

```
#ident "@(#)aliases    1.13    92/07/14 SMI"    /* SVr4.0 1.1
*/

##
#  Aliases can have any mix of upper and lower case on the left-
hand side,
#  but the right-hand side should be proper case (usually lower).
#
#  The program "newaliases" will need to be run after
#  NOTE   this file is updated for any changes to
#   show through to sendmail.
#
# @(#)aliases 1.8 86/07/16 SMI
##

# Following alias is required by the mail protocol, RFC 822.
# Set it to the address of a HUMAN who deals with this system's
mail problems.
Postmaster: root
audit_warn: cseisso, root

# Alias for mailer daemon; returned messages from our MAILER-
DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster

# Aliases to handle mail to programs or files, e.g., news or
vacation
```

```
# decode: "|/usr/bin/uudecode"
nobody: /dev/null
# Sample aliases:
# Alias for distribution list, members specified here:
#staff:wnj,mosher,sam,ecc,mckusick,sklower,olson,rwh@ernie
```

Topic: **AUDIT**
SubTopic:

```
# Alias for distribution list, members specified elsewhere:
#keyboards:  include:/usr/jfarrell/keyboards.list

# Alias for a person, so they can receive mail by several names:
#epa:eric

#######################
# Local aliases below #
#######################
```

Comments:

Whenever the audit daemon encounters an unusual condition while writing audit records, it invokes the /etc/security/audit_warn script. This script can be customized by individual sites to warn of conditions that might require manual intervention, or to handle them automatically. For all error conditions audit_warn writes a message to the console and sends a message to the auidt_warn alias.

Topic: AUDIT
SubTopic:

Objective 23

Verify the system provides an auditing function capable of accepting application level audit logging requests and a standard audit format is provided for use in application level auditing.

Rationale:

DII COE SRS Requirement:

3.2.2.8 The COE shall provide an auditing function capable of accepting application level audit logging requests.
3.2.2.8.1 The COE shall provide a standard audit format (e.g., syslog format) for use in application level auditing.

Test Actions:

Step: 1

Required Action:

Type the following command:

```
ps -eaf | grep syslog
```

Expected Results:

Output on the screen should resemble the following:

```
$ps -eaf | grep syslog
 root   161     1 53   Jul 29 ?        0:01 /usr/sbin/syslogd
 cisso  893   427  9 14:07:06 pts/2    0:00 grep syslog
$
```

Comments:

/etc/syslog.conf contains the configuration parameters for syslogd.

Topic: AUDIT
SubTopic: Protection of Audit Data

Objective 26

Verify the audit data is protected from change or deletion by general users.

Rationale:

DII COE SRS Requirement:

3.2.2.1.2 The audit function shall be protected from change or deletion by general users.

Test Actions:

Step: 1

Required Action:

As root, determine the name of the audit files listed in the /etc/security/audit_control file. For each "filename", as a NON-privileged user, type the following commands:

```
    vi "filename"
    rm "filename"
```

Expected Results:

Each command should cause an error message to be returned. Every audit filesystem listed in a line starting with "dir:" in the /etc/security/audit_control file should be accessible only to security administrators.

Comments:

Topic: Availability
SubTopic:

Objective 51
Verify the system provides the capability to perform system and database backups on a periodic basis.

Rationale:

DII COE SRS Requirement:
3.2.3.4 The COE shall provide the capability to perform system and database backups on a periodic basis.

Test Actions:
Step: 1

Required Action:
View the files contained in the /var/spool/cron/crontabs directory to determine whether the system is backed up automatically on a scheduled basis. From the system logbook or the System Administrator determine when the last system backup was performed and if backups are regularly performed. Determine if the backup tapes were labeled correctly.

Expected Results:
Backups are regularly performed either by cron jobs or by operational procedures.

Comments:
Cron jobs executed are logged in the file /var/cron/log.

Topic: **CRON JOBS**
SubTopic: **Permissions**

Objective 129
Verify cron has been securely configured. Determine which form of cron is used on the system (see rationale for cron forms).

Rationale:
UNIX has programs and systems that run automatically. Many of these systems require special privileges. If an attacker can compromise these systems, he may be able to gain direct unauthorized access to other parts of the operating system, or plan a back door to gain access at a later time.

There are three forms of crontab files. The oldest form has a line with a command to be executed whenever the time field is matched by the cron daemon. To execute the commands from this old-style crontab file as a user other than root, it is necessary to make the command listed in the crontab file use the su command.

The second form of the cron file has an extra field that indicates on whose behalf the command is being run.

The third form of cron protects directories with a separate crontab file for each user. The cron daemon examines all the files and dispatches jobs based on the user owning the file.

DII COE SRS Requirement:

Test Actions:
***Step:* 1**

Required Action:
Review the /etc/default/cron file to determine the PATH and SUPATH for cron jobs. The PATH variable is used for user jobs, the SUPATH variable for root jobs.

Expected Results:
Directories in the PATH and SUPATH and the files contained in these directories are not world or group writeable.

Comments:

The PATH and SUPATH determines where the system looks to find executables. The security implications of setting PATH and SUPATH should be carefully considered.

***Step:* 2**

Required Action:
Type in the following commands:

```
#ls -ldgb /var
#ls -ldgb /var/adm
#ls -ldgb /var/adm/cron
#ls -ldgb /var/spool
#ls -ldgb /var/spool/cron
#ls -ldgb /var/spool/cron/crontabs
#ls -ldgb /var/spool/cron/atjobs
```

```
#ls -ldgb /usr
#ls -ldgb /usr/lib
```

Topic: CRON JOBS
SubTopic: Permissions

Expected Results:
All directories are not world or group writeable.

Comments:

Step: 3

Required Action:
Type in the following commands:

```
#/bin/find /var/spool/cron/crontabs -type f -exec ls -ldb {} \; \
-exec /usr/ucb/more {} \;
#/bin/find /var/spool/cron/atjobs -type f -exec ls -ldb {} \; \ -
exec /usr/ucb/more {} \;
```

Expected Results:
All user crontab files are owned by the correct user and group, all files that are referenced in a users crontab file, or that are referenced by files in the crontab file are not world or group writeable, and the cron job tasks are appropriate.

Comments:
Ensure root cron job files do NOT source any other files not owned by root or which are group or world writeable. This is done by TIGER and maybe COPS and SPI.

Step: 4

Required Action:
Perform an `ls -ldg` and more on each file referenced in each crontab file to verify that none of the files are world writeable (check directories in the path of the referenced files also).

Expected Results:
All files that are referenced in the crontab file, or that are referenced by files in the crontab file are not world or group writeable and contain valid entries.

Comments:

Step: 5

Required Action:
Type in the following commands:

```
#ls -ldb /var/cron
#ls -ldb /var/cron/log
#/usr/ucb/more /var/cron/log
```

Expected Results:
The cron log directory and the cron log are not world or group writeable, and the cron jobs logged have been approved.

Comments:

Topic: CRON JOBS
SubTopic: Permissions

Step: 6

Required Action:
Type in the following commands:

```
#ls -ldb /var/adm/cron/cron.allow
#/usr/ucb/more /var/adm/cron/cron.allow
#ls -ldb /var/adm/cron/cron.deny
#/usr/ucb/more /var/adm/cron/cron.deny
```

Expected Results:
The cron.allow and cron.deny files are owned by root, are NOT world writeable, and contain the correct entries.

Comments:

Topic: **DAC**
SubTopic:

Objective 54
Verify Administration Tool use is limited to appropriate users.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:
Execute the following command:

```
ls -ldb /usr/bin/admintool
```

Expected Results:
The following permissions are displayed:

```
-r-xr-x--- bin bin /usr/bin/admintool
```

Comments:

Step: 2

Required Action:
Execute the following command:

```
#grep '^group' /etc/nsswitch.conf
```

Expected Results:
The group entry in nsswitch.conf will appear. The entry should be in one of the following forms:

```
group:  files nisplus
        or
group:  files
        or
group:  nisplus
```

Comments:
The Administration Tool permissions are granted to users who are members of the sysadmin group. This means that a user performing a task that modifies administration data on a system using the Administration Tool must be a member of the sysadmin group on the system where the task is being executed.

In the case of the Administration Tool, the /etc/group is searched for an entry for the sysadmin group (GID=14). If the entry exists, it uses the information listed there, and does not check the NIS+ group table.

Topic: **DAC**
SubTopic:

Step: 3

Required Action:

If the nsswitch.conf entry for group is of the form:

```
   group:   nisplus
```

then execute the following command:

```
  #  niscat group.org_dir | grep '^sysadmin'
```

Otherwise, execute the following command:

```
  #  grep '^sysadmin' /etc/group
```

Expected Results:

Only users authorized to execute the Administration Tool "admintool" should be members of the sysadmin

Comments:

The Administration Tool permissions are granted to users who are members of the sysadmin group. This means that a user performing a task that modifies administration data on a system using the Administration Tool must be a member of the sysadmin group on the system where the task is being executed. In the case of the Administration Tool, the /etc/group file is searched for an entry for the sysadmin group (GID=14). If the entry exists, it uses the information listed there, and does not check the NIS+ group table.

Topic: **DAC**
SubTopic:

Objective 55

Verify the privileged user’s account (e.g., root) and anything owned by that user is configured securely.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

Type the following command:

```
find / -user root ! -group bin -type f \
      ( -perm -2 -o -perm -20 \) \
       -exec ls -ldb {} \;
```

Expected Results:

Permissions are such that no user is able to write to any file especially executable and SUID, SGID files.

Comments:

Step: ***2***

Required Action:

Type the following command:

```
find / -user root ! -group bin ! -group sys -type d \
       \( -perm -2 -o -perm -20 \) -exec ls -ldb {} \;
```

Expected Results:

Permissions are such that no user is able to write to any directory that should not be written to.

Comments:

Topic: **DAC**
SubTopic:

Objective 56
Verify use of privileged commands (e.g., su) is logged and that a message is displayed on the console.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Type the following command:

```
   vi /etc/default/su
```

Expected Results:
The following lines should be uncommented:

(i.e., should not have a "#" in front of them)
SULOG=/var/adm/sulog
CONSOLE=/dev/console

Comments:
Entries in the file /etc/default/su determine the default conditions of the su command. The following entry enables a log of each time the su command is used to change to another user:

SULOG=/var/adm/sulog

(Security, Performance, and Accounting Administration)

A record of every time the su command is used, who uses it, and when it is made in the log file, /var/adm/sulog, enabling the system administrator to track who is using the superuser account.

The following entry enables a display on the console each time an attempt is made to use the su command to gain root access from a remote system.

CONSOLE=/dev/console

Topic: **DAC**
SubTopic:

Objective 57

Verify the access control information for the device maps is appropriate for each physical device.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

Review the following file:

```
/etc/security/device_maps
```

Expected Results:

In the file /etc/security/device_maps, only the device special files delivered with Solaris 2.4 are identified for each physical device.

Comments:

NOTE: This step cannot be performed if BSM is not installed. If BSM has been enabled, the device_maps file contains access control information about each physical device. Each device is represented by a one line entry of the form:

```
device-name:device-type:device-list
```

where

device-name is an arbitrary ASCII string naming the physical device.
device-type is an arbitrary ASCII string naming the generic device type.
device-list is a list of the device special files associated with the physical device. This field contains valid device special file path names separated by white space.

Topic: **DAC**
SubTopic:

Objective 59

Verify the lock out function is available for users to manually lock their terminals and users are required to re-authenticate themselves to unlock a locked terminal.

Rationale:

DII COE SRS Requirement:

3.2.4.12.4 The lock out function shall be available for users to manually invoke.
3.2.4.12.5 Users shall be required to re-authenticate themselves to unlock a locked terminal.

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
#xlock
```

Expected Results:

Screensaver appears.

Comments:

If the command results in a "xlock: not found" error, check for the presence of xlock on the system using the following command:

```
#find / -name "*xlock* -print
```

Step: 2

Required Action:

Press the Enter key and enter the Password.

Expected Results:

The password entry prompt appears and the screen unlocks.

Comments:

Step: 3

Required Action:

On DII COE Computers, click on the padlock symbol on the status bar at the bottom of the screen.

Expected Results:

Screensaver appears.

Comments:

Topic: **DAC**
SubTopic:

Objective 62

Verify the system protects objects from unauthorized access and is capable of including or excluding access to each object on a per user and on a per group basis.

Rationale:

DII COE SRS Requirement:

3.2.4.5 The COE shall, either by explicit user action or by default, protect objects from unauthorized access.
3.2.4.6 The COE shall be capable of including or excluding access to each object on a per user and on a per group basis.

Test Actions:

Step: 1

Required Action:

Administrator/Superuser logs into the host and assumes root.

Expected Results:

The host system prompt is displayed on the screen.

Comments:

Step: 2

Required Action:

Administrator/Superuser edits the contents of the /etc/shadow file.

Expected Results:

Contents of the /etc/shadow file are displayed on the screen. Administrator/Superuser is able to edit the file.

Comments:

Step: 3

Required Action:

Administrator/Superuser edits the contents of the /etc/group file.

Expected Results:

Contents of the /etc/group file are displayed on the screen. Administrator/Superuser is able to edit the file.

Comments:

Step: 4

Required Action:

Administrator/Superuser displays the file creation mask (umask) for his account. Type the command 'umask'.

Expected Results:

The file creation mask (umask) for his account is set to 77 (owner is given read, write, and execute privilege; group is given no privilege, and world is given no privilege).

Comments:

Topic: **DAC**
SubTopic:*:*

***Step:* 5**

Required Action:

Test default umask by creating new account and verifying correct DAC permissions. As root, run the admin tool application, and create a new account called test2. Set the home directory to /home/test2.

Expected Results:

The admin creates the account.

Comments

***Step:* 6**

Required Action:

Login to the host as the newly created test2 account. Type the command 'telnet localhost' and login as test2.

Expected Results:

The user ends up logged in as test2 within the window.

Comments:

***Step:* 7**

Required Action:

Verify the test2 account has the proper umask for correct DAC permissions. Type the command 'umask'.

Expected Results:

The system should display the result as '77'.

Comments:

***Step:* 8**

Required Action:

Administrator/Superuser logs out of the host.

Expected Results:

The host login prompt is displayed on the screen.

Comments:

***Step:* 9**

Required Action:

test1 logs into the host.

Expected Results:

Open Windows is started and three host windows are opened on the screen.

Comments:

Topic: **DAC**
SubTopic:

Objective 63
Verify the system is capable of restricting access to objects based on the user's identity and on access modes (e.g., read, write, execute).

Rationale:

DII COE SRS Requirement:
3.2.4.2 The COE shall restrict access to objects based on the user's identity and on access modes (e.g., read, write, execute).

Test Actions:
Step: 1

Required Action:
As unprivileged user1, execute the following commands:

```
user1>echo ls -CFA > /tmp/file1
user1>chmod 700 /tmp/file1
user1>ls -ld /tmp/file1
user1>/usr/ucb/more /tmp/file1
```

Expected Results:
Output will look similar to the following:

```
user1>ls -ld /tmp/file1
-rwx------  1 user1             8 Oct 17 16:49 /tmp/file1*
user1>/usr/ucb/more /tmp/file1
ls -CFA
```

Comments:

Step: 2

Required Action:
As unprivileged user2 (a member of the same group), execute the following commands:

```
user2>ls  /tmp/file1
user2>/usr/ucb/more /tmp/file1
user2>echo date > /tmp/file1
user2>/tmp/file1
```

Expected Results:
Output similar to the following will be produced:

```
user2>ls -ld /tmp/file1
-rwx------  1 user1             8 Oct 17 16:49 /tmp/file1*
user2>more /tmp/file1
/tmp/file1: Permission denied
user2>echo date > /tmp/file1
/tmp/file1: Permission denied
user2>/tmp/file1
/tmp/file1: Permission denied
user2>
```

Topic: **DAC**
SubTopic:

Comments:

Step: 3

Required Action:

As unprivileged user1, execute the following commands:

```
user1>chmod 750
user1>ls  /tmp/file1
user1>/usr/ucb/more /tmp/file1
```

Expected Results:

Output will look similar to the following:

```
user1>ls -ld /tmp/file1
-rwxr-x---  1 user1             8 Oct 17 16:49 /tmp/file1*
user1>/usr/ucb/more /tmp/file1
ls -CFA
```

Comments:

Step: 4

Required Action:

As unprivileged user2 (a member of the same group), execute the following commands:

```
user2>ls  /tmp/file1
user2>/usr/ucb/more /tmp/file1
user2>echo date > /tmp/file1
user2>/tmp/file1
```

Expected Results:

Output will look similar to the following:

```
user2>ls -ld /tmp/file1
-rwxrwx---  1 user1             8 Oct 17 16:49 /tmp/file1*
user2>/usr/ucb/more /tmp/file1
ls -CFA
user2>echo date > /tmp/file1
/tmp/file1: Permission denied
user2>/tmp/file1
file1       file2          file3         file4        file5
file6       file7          file8         file9        file10
```

Comments:

Step: 5

Required Action:

As unprivileged user1, execute the following commands:

```
user1>chmod 770
user1>ls  /tmp/file1
user1>/usr/ucb/more /tmp/file1
```

Expected Results:

Output will look similar to the following:

```
user1>ls -ld /tmp/file1
-rwxrwx---  1 user1             8 Oct 17 16:49 /tmp/file1*
user1>/usr/ucb/more /tmp/file1
ls -CFA
```

Comments:

Step: ***6***

Required Action:

As unprivileged user2 (a member of the same group), execute the following commands:

```
user2>ls  -ld /tmp/file1
user2>/usr/ucb/more /tmp/file1
user2>echo date > /tmp/file1
user2>/usr/ucb/more /tmp/file1
user2>/tmp/file1
```

Expected Results:

Output will look similar to the following:

```
user2>ls -ld /tmp/file1
-rwxrwx---  1 user1             8 Oct 17 16:49 /tmp/file1*
user2>/usr/ucb/more /tmp/file1
ls -CFA
user2>echo date > /tmp/file1
user2>/usr/ucb/more /tmp/file1
date
user2>/tmp/file1
Thu Oct 17 17:37:06 EDT 1996
```

Comments:

Step: ***7***

Required Action:

As unprivileged user1, execute the following commands:

```
user1>rm /tmp/file1

user1>ls /tmp/file1
```

Expected Results:

Output will look similar to the following:

```
user1>rm /tmp/file1
user1>ls -ld /tmp/file1
/tmp/file1: No such file or directory
```

Comments:

Topic: DAC
SubTopic: Permissions

Objective 53
Verify System Administration Tools are configured securely.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Verify that the admin security level is NOT set to level 0 - Browse the following file:

```
/etc/inetd.conf
```

Expected Results:
The admind entry does not specify security level 0(i.e., the string "-S 0" does not appear in the admind entry).

Comments:
The Administration Tool uses the distributed administration framework daemon (admind) to carry out the security tasks. The admind daemon process executes the request on the server on behalf of the client process.

Each request contains a set of credentials with a user ID (UID) and a set of group IDs (GIDs) to which the UID belongs. The server uses these credentials to perform identity and permission checks. Three levels of authentication security are available:

- Level 0 (AUTH_NONE) - No identity checking is done. All user IDs are set to the nobody identity. This level is used mostly for testing.

- Level 1 (AUTH_SYS) - The server accepts the original user and group identities directly from the client system and uses them as the identities for the authorization checks. The server does not check that the UID of the client represents the same user on the server system. It is assumed the administrator has made the user IDs and group IDs consistent on all systems in the network. Checks are made to see if the client has permission to execute the request.

- Level 2 (AUTH_DES) - Credentials are validated using DES authentication, and the server checks that the client has permission to execute the request. The user and group identities are obtained from databases on the server system by mapping the user's DES network identity (the DES entry in the NIS+ Cred table, for example) to a local UID and set of GIDs. The database used depends on which name service is selected on the server system. This level provides the most secure environment for performing administrative tasks and requires that a publickey entry exist for all server systems where the admind daemon is running, and for all users accessing the tools.

The Administration Tool uses the Level 1 authentication as the default. The security can be tightened to require Level 2 security checks by editing the /etc/inetd.conf file on each system and adding the -S 2 option to the admind entry. The servers on the domain must be set up to use DES security.

Topic: DAC
SubTopic: Orange Book Requirements (DAC)

Objective 270
Verify that the Operating System was designed to satisfy the C2 level of trust as defined in the "Orange Book".

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Produce the System and Network Administration manual for Solaris; turn to Appendix D and verify that Solaris was designed to meet the Discretionary Access Control requirements of the C2 level of trust as defined in the "Orange Book."

Expected Results:
The System and Network Administration manual shows that Solaris was designed to meet the requirements of the C2 level of trust as defined in the "Orange Book."

Comments:

Step: 2

Required Action:
Determine if formal certification has been received.

Expected Results:

Comments:

Topic: **FILE SYS SEC**
SubTopic:

Objective 81

Verify the shell used on the system resets the Internal Field Separator (IFS) variable when invoked.

Rationale:

The Internal Field Separator (IFS) variable can be set to indicate what characters separate input words. Most modern versions of the shell will reset their IFS value to a normal set of characters when invoked. Thus, shell files will behave properly. However, not all do (Garfinkel and Spafford, 1992).

Bourne shell inherits the value of its internal field separator from its environment. This can be used to obtain root access. In the Bourne shell, the IFS is the ASCII character used as a separator on the command line between command names and arguments. Normally the IFS is set to space or tab, but it can also be set by the user from environment variables. In UNIX, environmental variables are passed to child processes. The C library call popen(3) uses the Bourne shell and inherits the environment variables, including IFS. Because of this, the path passed to popen(3) can be altered so that an alternate program is executed. This means a setuid root program which uses popen(3) can be forced to run a program other than what it is intended to run.

If a root program does "popen("/bin/mail" ...)", and the IFS is set to " / ", then it runs the program "bin" with the command argument of "mail" and a userid of root. "/usr/lib/ex3.7preserve" is one of many programs you can use to exploit this. When "vi(1)" receives a hangup signal or when the command "p- reserve" is used, it executes the program "/usr/lib/ex3.7preserve", which preserves the current file you were editing and sends mail to you notifying you that your file was saved. To make certain it has permission to do this, "ex3.7 preserve" runs setuid to root. The security problem arises because when ex3.7 preserve tries to send mail to the user, it uses popen(3) to run "/bin/mail".

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

As an unprivileged user, insert the following text into a file named "ifs_test":

```
#!/bin/sh
# A test of the shell
cd /tmp
cat > tmp <<  E-O-F
echo "Security Vulnerability. Your shell does NOT reset the IFS
variable!"
E-O-F

cat > foo << E-O-F
echo "Your shell appears well behaved."
E-O-F

cat > test$$ <<E-O-F
```

```
/tmp/foo
E-O-F
```

```
chmod 700 tmp foo test$$

PATH=.:$PATH
IFS=/
export PATH IFS

test$$
rm -f tmp foo test$$SUID and SGID scripts should NEVER be used.

THEN execute the following commands:
chmod 700 ifs_test
ifs_test
```

Expected Results:

Script file exists.

Comments:

SUID and SGID scripts should NEVER be used.

Step: ***2***

Required Action:

As an unprivileged user, execute the following commands:

```
user1>chmod 700 ifs_test
user1>ifs_test
```

Expected Results:

Output other than "Your shell appears well behaved" indicates that the IFS variable does not get reset and under no condition should SUID or SGID scripts be used.

Comments:

SUID and SGID scripts should NEVER be used.

Step: ***3***

Required Action:

Attempt to exploit IFS by executing the following commands:

```
#  cat >~/bin/bin  #!/bin/sh  sh -i  ^D
#  chmod 755 ~/bin/bin
#  setenv IFS /
#  cd ~/bin
#  /usr/openwin/bin/loadmodule /sys/sun4c/OBJ/evqmod-sun4c.o
/etc/openwin/modules/evqload
# whoami
```

Expected Results:

The output should indicate that the user is NOT root.

Comments:

SUID and SGID scripts should NEVER be used.

Topic: **FILE SYS SEC**
SubTopic: **Expreserve**

Objective 83

Verify that the expreserve executable is secure.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following commands:

```
#showrev -p
#find / -name "*preserve*" -exec ls -ldb {} \;
```

Check to see if the expreserve executable is setuid root. If not, the following procedure won't work:

a. cd into to your home directory.

b. Create a file called "bin" containing the following lines:

```
#  (IFS= should be followed by a single space then return)
IFS=' '
cp /bin/sh /the/path/to/your/home/directory/xyzzy
chmod 4755 xyzzy
```

c. After saving the file (and exiting the editor) Type:

```
%  chmod 755 bin
%  /bin/sh
```

d. From this Bourne shell, type:

```
IFS=/ vi
```

e. You should be in vi. Type "a" (return) and then type a couple of lines of random text into the buffer.

f. Type: <Escape> :preserve

g. Next exit the editor usin the command:

```
        <Escape>  :wq
```

h. Enter the command:

```
        %  ls -l xyzzy
```

Topic: FILE SYS SEC
SubTopic: Expreserve

Expected Results:
The expreserve patch (ID = 102756-01) has been installed or the date shown for the file is after July 1993. This patch is not on the "Sun recommended" patchlist.

There should not be a setuid root Bourne shell in your home directory. If the ex command ":preserve" fails, instead you can run a shell from within vi with the command ":shell", from the shell get the pid of the editor and kill it with a hangup signal.

Comments:
Removal of executable permission will protect the system from this vulnerability, but will also mean that users who edit their files with either vi(1) or ex(1) and have their sessions interrupted, will not be able to recover their lost work. If you implement the above workaround, please advise your users to regularly save their editing sessions.

Step: 2

Required Action:
Check to see if the expreserve executable is setuid root. If not, the following procedure won't work:

a. cd into the test working directory.

b. Create a file called "bin" containing the following lines:

```
#  (IFS= should be followed by a single space then return)
IFS=‘ ‘
cp /bin/sh ./xyzzy
chmod 4755 ./xyzzy
```

c. After saving the file (and exiting the editor) Type:

```
%  chmod 755 bin
%  /bin/sh
```

d. From this Bourne shell, type:

```
IFS=/ vi
```

e. You should be in vi. Type "a" (return) and then type a couple of lines of random text into the buffer.

f. Type: `<Escape>` :preserve

g. Next exit the editor usin the command:

```
        <Escape>  :wq
```

h. Enter the command:

```
%  ls -l xyzzy
```

Topic: FILE SYS SEC
SubTopic: Expreserve

Expected Results:

There should not be a setuid root Bourne shell in the test working directory. If the ex command ":preserve" fails, instead you can run a shell from within vi with the command ":shell", from the shell get the pid of the editor and kill it with a hangup signal.

Comments:

Removal of executable permission will protect the system from this vulnerability, but will also mean that users who edit their files with either vi(1) or ex(1) and have their sessions interrupted, will not be able to recover their lost work. If you implement the above workaround, please advise your users to regularly save their editing.

Topic: **FILE SYS SEC**
SubTopic: **IP source routing and IP forwarding**

Objective 80

Verify that ip forwarding and ip source routing has been disabled.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Execute the following command:

```
#vi /etc/rc2.d/S69inet
```

to determine if source routing and ip forwarding has been disabled.

Expected Results:

The lines below are uncommented:

```
ndd -set /dev/ip ip_forwarding 0
ndd -set /dev/ip ip_ip_forward_src_routed 0
```

Comments:

Disable source routing and ip forwarding.

Topic: **FILE SYS SEC**
SubTopic: **Path**

Objective 87

Verify root's search path is correct.

Rationale:

A search path should never contain the current directory. This is especially true of the superuser account. More generally, a search path should never include a directory that is writeable by other users.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root execute the following commands:

```
#echo $PATH
```

OR

review the root search path found in the /.profile, /.cshrc, and /.login files.

Expected Results:

Root’s search path does not include the current directory (specified by a ".").

Comments:

Step: 2

Required Action:

As root, execute the following command:

```
ls -ldb `echo $PATH | sed 's/:/ /g'`
```

Expected Results:

None of the directories in the search path should be world writeable.

Comments:

Topic: **FILE SYS SEC**
SubTopic: **Permissions**

Objective 66

Ensure the file systems are configured correctly and securely.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Execute the following command and then review the output:

```
#  df -k
```

Expected Results:

The file system should be appropriately partitioned so that no filesystem is approaching 100% full.

Comments:

Topic: **FILE SYS SEC**
SubTopic: **Permissions**

Objective 70

Verify root’s startup files are only writeable by root.

Rationale:

Various programs have methods of automatic initialization to set options and variables for the user. All startup files should be protected so only the user can write to them. It is particularly important that the startup files the superuser uses are not writeable by others.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following commands from root’s home directory and verify from the output that the files listed are writeable only by root:

```
#ls -ldb /.login
#ls -ldb /.profile
#ls -ldb /etc/profile
#ls -ldb /.cshrc
#ls -ldb /.kshrc
#ls -ldb /.emacs
#ls -ldb /.exrc
#ls -ldb /.forward
#ls -ldb /.rhosts
#ls -ldb /.dtprofile
#ls -ldb /.Xdefaults
```

Expected Results:

Permissions of existing files is 600 or 400 and are owned by root.

Comments:

Depending on the system configuration, all of the files listed in "Required Actions" may not exist. Solaris 2.4 does not install with the listed files present. The existence of any one of these files indicates an addition of the file by the System Administrator.

Step: 2

Required Action:

As root, execute the following commands from root’s home directory:

```
#/usr/ucb/more  /.login
#/usr/ucb/more  /.profile
#/usr/ucbmore  /etc/profile
#/usr/ucb/more  /.cshrc
#/usr/ucb/more  /.kshrc
#/usr/ucb/more /.emacs
#/usr/ucb/more  /.exrc
#/usr/ucb/more  /.forward
#/usr/ucb/more  /.dtprofile
#/usr/ucb/more /.Xdefaults
and on any executable that is referenced in the file being viewed
execute the command:
```

```
ls -ldb
```

Topic:: ***FILE SYS SEC***
SubTopic: Permissions

Expected Results:
Permissions of all files referenced in the listed files are 600 or 400 and are owned by root.

Comments

Topic: **FILE SYS SEC**
SubTopic: **Permissions**

Objective 72

Verify all root executable files are owned by root and are not world or group writeable.

Rationale:

System Administrators should be trained to type in full pathname of files to be executed and to ensure that any executable that is not located in a protected directories are safe to execute.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following commands:

```
#ls -lgdb /etc /usr /usr/bin /usr/sbin /usr/5bin
```

Expected Results:

Listed directories are owned by root and are not world or group writeable.

Comments:

All executables run by root should be located in a directory where every directory in the path is owned by root and is not group or world writeable. In particular, the following directories should not be group or world writeable: /bin, /etc, /usr/sbin, /usr/bin, /usr/5bin, /usr/ucb. System Administrators should be trained to type in full pathname of files to be executed and to ensure that any executable that is not located in the protected directories listed above are safe to execute.

Step: 2

Required Action:

As root, execute the following commands:

```
#find /etc -user root \( -perm -2 -o -perm -20 \) \
         ! -type l -exec ls -lgdb {} \;
#find /usr/bin -user root \( -perm -2 -o -perm -20 \) \
         ! -type l -exec ls -lgdb {} \;
#find /usr/sbin -user root \( -perm -2 -o -perm -20 \) \
         ! -type l -exec ls -lgdb {} \;
#find /usr/5bin -user root \( -perm -2 -o -perm -20 \) \
         ! -type l -exec ls -lgdb {} \;
```

Expected Results:

There should be no files listed indicating that there are no world/group writeable root owned files.

Comments:

All executables run by root should be owned by root and all executables run by root should not be world or group writeable.

Topic: FILE SYS SEC
SubTopic: Permissions

Objective 78

Identify all world-writeable files on the system and verify their need for world-write access.

Rationale:

World-writeable files, directories, and devices represent a potential security hole in a system. It is important to periodically identify them and verify the need for world-write access. Notable files that may be world-writeable include: /tmp, /usr/tmp, and /dev/tty*(Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following commands:

```
# /bin/find / -type f \( -perm -2 -o -perm -20 \) \
-exec ls -lgb {} \;

# /bin/find / -type d \( -perm -2 -o -perm -20 \) \
-exec ls -lgdb {} \;
```

Expected Results:

There are no unexpected world writeable files or directories on your system. Files should be world-writeable only if there is a legitimate requirement.

COPS, Tiger, SPI all provide checking of file permissions.

Comments:

The following files and directories may safely remain world-writeable:

/tmp and contents
/var/tmp and contents
/var/preserve
/var/mail
(and many more...)

Topic: FILE SYS SEC
SubTopic: Permissions

Objective 79

Verify that all world-readable, but not world or group writeable, non-setuid/setgid system files and directories are owned by root. (see rationale)

Rationale:

Many systems ship files and directories owned by bin (or sys). This varies from system to system and may have serious security implications.

CHANGE all non-setuid files and all non-setgid files and directories that are world readable but not world or group writeable and that are owned by bin to ownership of root, with group id 0 (wheel group under SunOS 4.1.x).

Please note that under Solaris 2.x changing ownership of system files can cause warning messages during installation of patches and system packages - anything else should be verified with the vendor.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following command:

```
/usr/bin/find / -perm -4 ! \( -perm -6022 \)  \
       \( -type f  -o -type d \) \
       ! -user root -group 0 -exec ls -lgdb {} \;
```

Expected Results:

Any output from this command indicates a file or directory that does not meet the criteria listed in the rationale and should be investigated carefully.

Comments:

Use of a tool such as Tiger, COPS, or SPI would be very useful and save work.

Topic: FILE SYS SEC
SubTopic: Permissions

Objective 82

Verify the startup and shutdown scripts are valid and protected.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following command:

```
/bin/find /etc \( -perm -2 -o -perm -20 \) -exec ls -ld {}\;
```

Expected Results:

There should be no output indicating that the /etc directory and its contents are not group or world writeable.

Comments:

Step: 2

Required Action:

Review all startup and shutdown scripts and configuration files. These scripts are located in the /etc directory and all begin with rc.

Expected Results:

Any task performed in the startup script is performed securely. Any service started or task performed is approved. Any directory that contains a script, executable, or configuration file that is executed in the rc scripts during bootup and shutdown is not writeable by a user other than root.

Comments:

Work intensive!

Topic: FILE SYS SEC
SubTopic: Permissions

Objective 85
Identify the SUID and SGID files on the system and verify their need for SUID and SGID privilege.

Rationale:
SUID and SGID files allow an unprivileged user to accomplish tasks that require privileges. When a SUID program is run, its effective UID becomes that of the user who created the program, rather than the user who is running it. When a SGID program runs, its effect GID becomes that of the creating user.

Shell scripts that have the setuid or setgid bits set on them are not secure, regardless of how many safeguards are taken when writing them. Setuid and setgid shell scripts should never be allowed on any UNIX system (Curry, 1990).

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
As root, execute the following command:

```
#/bin/find / -type f \( -perm -4000 -o -perm -2000 \) \
-exec ls -lgdb {} \;
```

Expected Results:
Verify that all of the programs listed as output should be SUID or SGID. Only authorized files should be SUID or SGID.

Comments:
Depending on the system configuration, the output may be lengthy and it may be easier to review if piped to an output file, which can then be printed and reviewed. There are a large number (about 90) SUID and SGID programs that are installed as part of Solaris 2.4. Tools such as COPS, Tiger, and SPI report SUID and SGID programs.

Topic: FILE SYS SEC
SubTopic: Permissions

Objective 86
Determine if users can "give away" files, and if so, if they can "give away" an SUID file to root.

Rationale:
The last defense against system crackers are the permissions offered by the file system. Each file or directory has three sets of permission bits associated with it: one set for the user who owns the file, one set for the users in the group with which the file is associated, and one set for all other users (the "world" permissions). Each set contains three identical permission bits, which control the following (Curry, 1990):

read - If set, the file or directory may be read. In the case of a directory, read access allows a user to see the contents of a directory (the names of the files contained therein), but not to access them.

write - If set, the file or directory may be written (modified). In the case of a directory, write permission implies the ability to create, delete, and rename files. Note that the ability to remove a file is not controlled by the permissions on the file, but rather the permissions on the directory containing the file.

execute - If set, the file or directory may be executed (searched). In the case of a directory, execute permission implies the ability to access files contained in that directory.

In addition, a fourth permission bit is available in each set of permissions. This bit has a different meaning in each set of permission bits:

setuid - If set in the owner permissions, this bit controls the "set user id" (setuid) status of a file. Setuid status means that when a program is executed, it executes with the permissions of the user owning the program, in addition to the permission of the user executing the program. This bit is meaningless on nonexecutable files.

setgid - If set in the group permissions, this bit controls the "set group id" (setgid) status of a file. This behaves in exactly the same way as the setuid bit, except that the group id is affected instead. This bit is meaningless on non-executable files (but see below).

sticky - If set in the world permissions, the "sticky" bit tells the operating system to do special things with the text image of an executable file. It is mostly a hold-over from older versions of UNIX, and has little if any use today. This bit is also meaningless on nonexecutable files (but see below).

Under some versions of UNIX, users can run the chown command to change the ownership of a file that they own to that of any other user on the system, allowing them to "give away the file."

DII COE SRS Requirement:
3.2.4.4 The COE shall provide controls to limit the propagation of access rights.

Test Actions:

Topic: FILE SYS SEC
SubTopic: Permissions

Step: 1

Required Action:
As an unprivileged user, execute the following commands:

```
%touch test
%ls -lg test
%chown root test
%ls -lg test
%chmod 4755 test
%ls -lg test
%chown root test
%ls -lg test
```

Expected Results:
Each attempt to change the owner to root should result in an error message of "Permission denied". Output should be similar to the following:

```
user>touch test
user>ls -lg test
-rw-------  1 mls      rg021           0 Oct 21 09:30 test
user>chown root test
chown: test: Not owner
user>chmod 4755 test
user>ls -lg test
-rwsr-xr-x  1 mls      rg021           0 Oct 21 09:30 test*
user>chown root test
chown: test: Not owner
user>ls -lg test
-rwsr-xr-x  1 mls      rg021           0 Oct 21 09:30 test*
user>rm test
user>
```

Comments:
A general user should not be able to change the ownership of an SUID or SGID file (or any file) to any other user especially root.

Topic: FILE SYS SEC
SubTopic: Permissions

Step: ***2***

Required Action:
As an unprivileged user, execute the following commands:

```
%touch test
%ls -lg test
%chgrp root test
%ls -lg test
%chmod 2755 test
%ls -lg test
%chown root test
%ls -lg test
ser1>touch test
user1>ls -lg test
user1>chgrp root test
chgrp: test: Not owner
user1>ls -lg test
```

Expected Results:
Each attempt to change the group to root should result in an error message of "Permission denied". Output should be similar to the following:

```
rw-------   1 mls      rg021           0 Oct 21 09:41 test
-rw-------  1 mls      rg021           0 Oct 21 09:41 test
```

Comments:
A general user should not be able to change the group of an SUID or SGID file (or any file) to any other group especially root.

Topic: FILE SYS SEC
SubTopic: Unauthorized Device Files

Objective 75

Ensure no unauthorized device files are present on the system.

Rationale:

The system's disks should be periodically scanned for unauthorized device files.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following command:

```
#/bin/find / \( -type c -o -type b \) -exec ls -lgdb {} \; | grep
-v "/dev/" | grep -v "/devices/"
```

Expected Results:

There are no unexpected special files outside the /dev directory.

Comments:

Any device outside the /dev and /devices directory should be viewed with GREAT suspicion. NOTE: ncheck locates SUID files also. The -s parameter of the ncheck command displays special files and files with set-user-ID mode. This parameter can be used to discover concealed violations of security policy. The ncheck command would be run as root and executed as follows:

#/etc/ncheck -s

Step: 2

Required Action:

As root, execute the following command:

```
/bin/find /dev ! \( -type l -o -type c -o -type b \) \
-exec ls -lgdb {} \;
```

Expected Results:

All files in /dev and /devices are special files.

Comments:

Step: 3

Required Action:

As root, execute the following command:

```
#/bin/find / \( -type c -o -type b \) ! -user root \
-exec ls -ldb {} \;
```

Expected Results:

There are no special device files owned by root that should not be owned by root.

Comments:

Any device outside the /dev and /devices directory not owned by root should be viewed with even GREATER suspicion.

Topic: FINGER
SubTopic: Penetrate

Objective 130

Determine if finger and fingerd are enabled on the system. If enabled, verify Finger is securely configured.

Rationale:

The "finger" service, provided by the finger program, allows you to obtain information about a user such as her full name, home directory, last login time, and in some cases when she last received mail and/or read her mail. The fingerd program allows users on remote hosts to obtain this information (Curry, 1990).

A bug in fingerd was also exercised with success by the Internet worm. If your version of fingerd is older than November 5, 1988, it should be replaced with a newer version (Curry, 1990).

The finger program has two uses: If finger is run with no arguments, the program prints the username, full name, location, login time, and office telephone number of every user currently logged into the local system. If finger is run with a name argument, the program searches through the /etc/passwd file and prints detailed information for every user with a first, last, or user name that matches the name you specified. finger makes it easy for intruders to get a list of the users on the system.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
user1>finger root@localhost
```

Expected Results:

Error message indicates that the finger daemon is not enabled. Output of information regarding root indicates that finger is enabled.

Comments:

Finger should NOT be enabled unless there is a legitimate need for it.
Related services that should be considered for removal are systat and netstat.

Step: 2

Required Action:

Execute the following command:

```
user1>finger @localhost
```

Expected Results:

Only login information on users currently logged on the system are provided.

Comments:

There is a bug in some operating systems which allows a remote finger request to dump all known user finger profiles back out to the requester. The same hack in a different fashion on Solaris

4.1.x will give random users profile.

Topic FINGER
SubTopic: Penetrate

Step: 3

Required Action:
Execute the following command:

```
user1>finger 23234123123123123@localhost
```

Expected Results:

Comments:
There is a bug in some operating systems which allows a remote finger request to dump all known user finger profiles back out to the requester. The same hack in a different fashion on Solaris 4.1.x will give random users profiles.

Topic: FTP
SubTopic:

Objective 136

Verify the FTP users file contains the appropriate accounts.

Rationale:

The /etc/ftpusers file contains a list of the users who are not allowed to use FTP to access any files. This file should contain all accounts that are not used by actual users.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type the following commands:

```
ls -lg /etc/ftpusers
more /etc/ftpusers
```

Expected Results:

The permissions do not allow group/world write and the file is owned by root. Typical accounts that should be included are uucp, news, bin, ingress, news, nobody, daemon, and root.

Comments:

The ftpusers file should contain a list of users who are not allowed access to the system using the File Transfer Protocol (FTP). If this file is missing, the list of users is considered to be empty, so that any user may use FTP to access the system if the other criteria for access are met.

Topic: **FTP**

SubTopic:

Objective 138

Determine whether Trivial FTP is enabled on the system and if enabled, verify that it has been securely configured.

Rationale:

The TFTP is used to allow diskless hosts to boot from the network. Basically, TFTP is a stripped-down version of FTP - there is no user authentication. Because they are so stripped-down, many implementations of TFTP have security holes (Curry, 1990).

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

As an unprivileged user execute the following commands:

```
%  tftp
tftp>  connect localhost
tftp>  get /etc/passwd testfile
tftp>  quit
%ls -l testfile
%more testfile
%rm testfile
```

Expected Results:

If tftp does not respond with "File not found," and instead transfers the file, the version of tftp should be replaced with a newer one.

Comments:

The use of tftp does not require an account or password on the remote system. The -s options ensures that tftpd will only start with home directory and its root directory both /tftpboot.

Topic: **FTP**
SubTopic:

Objective 140
Verify that Trivial FTP does not run with privileges.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Type the following command:

```
% ls -lF /user/bin/tftp
```

Verify that the file is not running SUID or SGID.

Expected Results:
The tftp file should not have the SUID or SGID bits set.

Comments:

Topic: FTP
SubTopic: Anonymous FTP

Objective 134
Determine whether anonymous FTP is enabled on the system. If anonymous FTP is enabled, verify that it has been securely configured.

Rationale:
Anonymous FTP allows users who do not have an account on a machine to have restricted access in order to transfer from a specific directory. Because the anonymous FTP feature allows anyone to access the system (albeit in a very limited way), it should not be made available on every host on the network. If anonymous ftp is required, one machine should be chosen (preferably a server or standalone host) on which to allow this service. (Curry, 1990)

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
To ascertain whether you are running anonymous ftp, try to connect to the localhost using anonymous ftp. Be sure to give an RFC822-compliant username (e.g., mcguire@ncr.disa.mil) as the password. Type the following commands to ascertain whether anonymous ftp is enabled:

```
%  ftp <hostname>
name (localhost:idname):  anonymous
```

Expected Results:
If the error message "530 User anonymous unknown" is returned then anonymous ftp is disabled. If the system instead replies with the string "331 Guest login ok" and then prompts for a password, anonymous ftp access is enabled.

Comments:
Anonymous ftp should not be enabled unless there is a legitimate business need.

Step: 2

Required Action:
To determine if anonymous ftp is securely configured, verify that the ftp account has been created and has been disabled by placing an asterisk (*) in the password field. Verify that the account has been given a special home directory, such as /usr/ftp or /usr/spool/ftp.

Expected Results:

Comments:

Step: 3

Required Action:
Verify that the ftp owns its home directory and that it is unwriteable by anyone.

Expected Results:

Comments:

Topic: FTP

SubTopic: ***Anonymous FTP***

Step: 4

Required Action:

Verify that the directory ftp/bin is owned by the super-user and unwriteable by anyone. Verify that a copy of the ls program is in this directory.

Expected Results:

Comments:

Step: 5

Required Action:

Verify that the directory ftp/etc is owned by the super-user and unwriteable by anyone. Verify that copies of the password and group files are in this directory, with all the password fields changed to asterisks (*). Note that the only account that must be present is "ftp."

Verify the directory ftp/pub is owned by "ftp" and world-writeable.

Expected Results:

Comments:

Step: 6

Required Action:

Verify that the directory ftp/pub is owned by "ftp" and world-writeable.

Expected Results:

Comments:

Topic: FTP
SubTopic: Penetration Test

Objective 137

Check for an early FTP bug that allows user login as root.

Rationale:

While looking at ftp, one should check for an older bug that was once widely exploited.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

From a networked host, type the following commands to check for an early FTP bug:

```
%  ftp -n
ftp> open <localhost>
ftp>  quote user ftp
ftp>  quote pass ftp
```

Expected Results:

If the bug is not fixed, the user will now be logged in as root.

Comments:

The ftp bug should be fixed.

Topic: HARDWARE/FIRMWARE
SubTopic:

Objective 184
Verify the single user boot or system firmware password is set, and the system is configured such that a password must be entered to boot to a single-user state.

Rationale:

DII COE SRS Requirement:
3.2.12.3 The COE shall be configured such that a password must be entered to boot to a single-user state.

Test Actions:
Step: 1

Required Action:
Type the following command:

```
#eeprom security-mode
```

Expected Results:
The EEPROM configuration parameters are set to a security-mode other than none (Preferably Full) as shown below.

```
# eeprom security-mode
security-mode=full
```

Comments:
The eeprom command displays or changes the values of parameters in the EEPROM. It processes parameters in the order given. When processing a parameter accompanied by a value, eeprom makes the indicated alteration to the EEPROM; otherwise it displays the parameter's value. When given no parameter specifiers, eeprom displays the values of all EEPROM parameters. Only the super-user may alter the EEPROM contents.

The following EEPROM parameters have security significance:

- security-#badlogins Contains the number of incorrect security password attempts to the firmware.

- security-mode Contains the firmware security level (options: none, command, or full). If set to command or
full, the system will prompt the user for a PROM security password. The default setting is none.

- security-password Contains the firmware security password (never displayed). The password can be set only
 when the security-mode is set to command or full.

Topic: **HARDWARE/FIRMWARE**
SubTopic:

Step: ***2***

Required Action:
Halt the system. When the machine is halted, attempt to reboot into single user mode with the following command:

```
>boot <disk> -s
```

OR depending on the machine architecture

```
>b <disk> -s
```

Expected Results:
The user should be challenged for the eeprom password when booting into single-user mode.

Comments:

Topic: I&A
SubTopic:

Objective 97
Verify I&A mechanisms are configured for secure operation.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Enter a valid user ID and invalid password at the login prompt.

Expected Results:
"Login incorrect" message is displayed on the screen. The host login prompt is redisplayed on the screen.

Comments:

Step: 2

Required Action:
Enter an invalid user ID and valid password at the login prompt.

Expected Results:
"Login incorrect" message is displayed on the screen. The login prompt is redisplayed on the screen.

Comments:

Step: 3

Required Action:
Enter an invalid user ID and invalid password at the login prompt.

Expected Results:
"Login incorrect" message is displayed on the screen. The login prompt is redisplayed on the screen.

Comments:

Step: 4

Required Action:
Attempt two additional invalid logins.

Expected Results:
A "Login incorrect" message is displayed on the screen after each invalid login attempt. After the final attempt, the "REPEATED LOGIN FAILURES" message is displayed on the screen. (This message may take several minutes to display). Note this information is logged as well to the file /var/adm/messages.

Topic: **I&A**
SubTopic:

Comments:
If a window manager is not running, the message will be logged to /var/adm/loginlog if that file has been created.

Step: 5
Required Action:
Attempt to log in as root.

Expected Results:
The "NOT ON SYSTEM CONSOLE" message is displayed on the screen.

Comments:

Step: 6
Required Action:
The test account supplies valid user ID and valid password at the login prompt.

Expected Results:
The user is logged into the host.

Comments:

Step: 7
Required Action:
The test account logs out of the host.

Expected Results:
The host login prompt is displayed on the screen.

Comments:

Step: 8
Required Action:
Administrator/Superuser logs into the host and assumes root by using the su to root command.

Expected Results:
The host system prompt is displayed on the screen.

Comments:

Step: 9
Required Action:
As an Administrator/Superuser display user authentication data in the /etc/shadow file using the following command:

```
#/usr/ucb/more /etc/shadow
```

Expected Results:

User identification and authentication data is displayed on the screen. Users are uniquely identified and passwords are encrypted.

Topic: **I&A**

SubTopic:

Comments:
Not all users are in the /etc/shadow due to NIS/NIS+

Step: ***10***

Required Action:
As an Administrator/Superuser display the permissions for the /etc/shadow file using the following command:

```
ls - ld /etc/shadow
```

Expected Results:

Permissions for the /etc/security/passwd.adjunct file are 600 and the owner is root showing that access to this file is limited to the owner (root)

Comments:

Step: ***11***

Required Action:
Administrator/Superuser logs out of the host.

Expected Results:
The host login prompt is displayed on the screen.

Comments:

Topic: **I&A**
SubTopic:

Objective 107

Verify the system prohibits direct login as a trusted user (e.g., root). Also verify the system requires trusted users to change their effective userID to gain access to root (e.g., su) and to reauthenticate before requesting access to privileged fns.

Rationale:

DII COE SRS Requirement:

3.2.1.1.2 The COE shall prohibit direct login as a trusted user (e.g., the root user, or super user, etc.).
3.2.1.1.3 The COE shall provide the capability for trusted users to gain access to root through a process of changing their effective user identifier (userID) (e.g., su to root).
3.2.1.1.4 The COE shall require trusted users to re-authenticate before requesting access to functions that require system privileges.

Test Actions:

Step: 1

Required Action:

Browse the following file:

```
/etc/default/login
```

Expected Results:

The following line should be uncommented in the /etc/default/login file:

```
CONSOLE=/dev/console
```

This ensures that root can only log in at the system console, not from any remote terminal.

The file /etc/default/login is owned by root.

The file /etc/default/login has permissions 644.

Comments:

An entry in the file /etc/default/login determines the root access restrictions. If the following command appears in the file, then root access is restricted to the console:

CONSOLE=/dev/console

Any user who tries to remotely log into the system must first login to his account, and then use the su command to become root. (Security, Performance, and Accounting Administration)

Topic: **I&A**
SubTopic: **Accounts**

Objective 98

Verify there are no accounts on the system that have not been used within a reasonable amount of time.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in and run the following script to determine which users have not logged in within the last month:

```
#!/bin/sh
date
uname -a
PATH=/bin:/usr/bin;export PATH
umask 077
THIS_MONTH=`date | awk '{print $2}'`
/bin/last | /bin/grep $THIS_MONTH | awk '{print $1}' | sort -u >
users1$$
cat /etc/passwd | /bin/awk -F: '{print $1}' | /bin/sort -u >
users2$$
/bin/comm -13 users[12]$$
/bin/rm -f users[12]$$
```

Expected Results:

No USER login account names should be returned. If any user names are returned these should be considered dormant accounts and should be disabled or deleted.

Comments:

Topic: I&A
SubTopic: Accounts

Objective 100
Verify there are no duplicate GIDs.

Rationale:
The UNIX operating system relies on the GID to identify groups. It does not rely on the group name. Therefore, if two different group names have the same GID they are indistinguishable to the operating system.

DII COE SRS Requirement:

Test Actions:
Step: 1
Required Action:
If running NIS execute the following command:

```
# /bin/niscat group.org_dir
```

OR if NOT running NIS execute the following command:

```
#/usr/bin/more /etc/group
```

Verify there are no duplicate GIDS and that appropriate users belong to the system groups.

Expected Results:
There should not be duplicate GIDs.

Comments:
Group ids must be distinct integers between 0 and 32,767. If the environment is networked, users should have the same unique UID across the entire network. GID 0 is generally reserved for the groups "root" or "wheel" and GID 1 is reserved for the group "daemon".

If the RUNNING NIS command is used on a non NIS running machine, the following output is produced:

```
# /bin/niscat passwd.org_dir
passwd.org_dir: NIS+ servers unreachable.
#
```

Topic: I&A
SubTopic: Password Management

Objective 1

Verify all passwords transferred across the network are protected.

Rationale:

DII COE SRS Requirement:

3.2.1.6 If a COE component transfers a user's password across a network to another COE component, the password shall be protected.

Test Actions:

Step: 1

Required Action:

The only real method of testing this is through the use of a sniffer!!!

Expected Results:

Comments:

Topic: I&A
SubTopic: Password Management

Objective 105
Verify the system enforces individual user accountability, a globally-unique valid userid and password is required for all users to access the system, and the user's identity is associated with all auditable actions

Rationale:
Some sites have installed accounts with names such as "who," "date," "lpq," and so on that execute simple commands. These accounts are intended to allow users to execute these commands without having to log in to the machine. Typically these accounts have no password associated with them, and can thus be used by anyone. Many of the accounts are given a user id of zero, so that they execute with super-user permissions (Curry, 1990).

The problem with these accounts is that they open potential security holes. By not having passwords on them, and by having super-user permissions, these accounts practically invite crackers to try to penetrate them. Usually, if the cracker can gain access to the system, penetrating these accounts is simple, because each account executes a different command. If the cracker can replace any one of these commands with one of his own, he can then use the unprotected account to execute his program with super-user permissions (Curry, 1990).

Simply put, accounts without passwords should not be allowed on any UNIX system (Curry, 1990).

DII COE SRS Requirement:
3.2.1.1 The COE shall enforce individual accountability by providing the capability to uniquely identify each individual system user.
3.2.1.1.1 The COE shall require users to identify themselves before beginning to perform any actions that the system is expected to mediate.
3.2.1.2 Each user shall be identified by a globally unique user name or userID that will follow a standard set of processes or rules for formation.
3.2.1.3 The COE shall provide the capability of associating the user’s identity with all auditable actions taken by that individual.

Test Actions:
Step: 1

Required Action:
As root execute the following command:

```
# logins -p
```

Expected Results:
There should be no output from this command. This indicates that all accounts have passwords.

NOTE: If a password of <return> is assigned by root, this test does not work as the password field in /etc/shadow contains a value for the password. The only remedy for this is a dictionary search.

Comments:
The logins -p command provides a list of login accounts that have no passwords. The output of

this command can be used to make sure that all users on the system have a password.

Topic: **I&A**
SubTopic: **Password Management**

Objective 106

Verify the installation-provided userIDs do not have default passwords.

Rationale:

Several accounts come with a UNIX computer system. (These accounts are normally at the beginning of the /etc/passwd file and have names like bin, lib, uucp, and news.) Either disable these accounts or change their passwords (Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Attempt to log into each of the following IDs with its default password:

ID	Password
guest	guest
root	root
system	manager

Expected Results:

The default passwords should not be valid for the accounts.

Comments:

After installation be sure to change all default passwords, lock the account, or delete the account.

COPS, Tiger, and SPI check for common default passwords.

Topic: I&A
SubTopic: Password Management

Objective 112

Verify password life is limited to a maximum of 180 days and the user is notified prior to password expiration.

Rationale:

Some UNIX systems allow the system administrator to set a "lifetime" for passwords. Users whose passwords are older than the time allowed are forced to change their passwords the next time they log in. If a user's password is exceptionally old, the system may prevent the user from logging in altogether (Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

3.2.1.4.2 Password life shall be limited to a maximum of 180 days. The COE shall notify the user prior to password expiration.

Test Actions:

Step: 1

Required Action:

As root execute the following command:

```
#passwd -s -a
```

Expected Results:

All accounts should have an appropriate maximum password lifetime.

Comments:

To display information about passwords, the passwd -s -a command can be used. The following pieces of information are provided:

- Login name
- Password status (NP if no password, LK if login is locked, or PS for anything else)
- Date the password was last changed
- Minimum number of days after the last password change before the user can change the password
- Maximum number of days between password changes
- Number of warning days before the password must be changed

Topic: I&A
SubTopic: Orange Book Requirements (I&A)

Objective 271

Verify that the Operating System was designed to satisfy the C2 level of trust as defined in the "Orange Book".

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Review Solaris SHIELD Basic Security Manual, Chapter 5; turn to the appropriate section(s) which demonstrate the ability of the NMS to satisfy the "Orange Book" requirements.

Expected Results:

Section(s) are present in the manual which verify that the component Operating System was designed to meet the C2 requirements of the "Orange Book."

Comments:

Step: 2

Required Action:

Determine if formal certification has been received.

Expected Results:

Documentation indicates that formal certification has been given.

Comments:

Topic: **I&A**
SubTopic: **Accounts**

Objective 99

Verify there are no duplicate UIDs.

Rationale:

The UNIX operating system relies on the UID to identify accounts. It does not rely on the account name. Therefore, if two different account names have the same UID they are indistinguishable to the operating system.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

If running NIS, execute the following command:

```
#/bin/niscat passwd.org_dir
```

OR if NOT running NIS execute the following command:

```
#/usr/bin/more /etc/passwd
```

Verify that there are no duplicate UIDs.

Expected Results:

There should not be duplicate UIDs. If there are duplicate UIDs, the accounts should be disabled.

Comments:

User ids must be distinct integers between 0 and 32,767. If the environment is networked, users should have the same unique UID across the entire network. Root uses UID 0, Bin uses UID 1, and Daemon uses UID 2. In addition, it is customary to use the lower UIDs for non-human logins (i.e., UUCP). It is not recommended to re-use UIDs after a user account is deleted.

Topic: **I&A**
SubTopic: **Accounts**

Objective 103

Verify site identifying information is stored for all user accounts on the system.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

If NOT running NIS, browse the /etc/passwd file using the following command:

```
#/usr/bin/vi /etc/passwd
```

OR if RUNNING NIS, use the following command:

```
#/bin/niscat passwd.org_dir
```

Expected Results:

The fifth field should be filled in with relevant data (i.e., full user name and user location).

Comments:

If the RUNNING NIS command is used on a non NIS running machine, the following output is produced:

```
# /bin/niscat passwd.org_dir
passwd.org_dir: NIS+ servers unreachable.
#
```

Topic: I&A
SubTopic: Accounts

Objective 102
Verify there are no guest accounts on the system.

Rationale:
Guest accounts present a security hole. By their nature, these accounts are rarely used, some are always used by people who should only have access to the machine for the short period of time that they are guests. The most secure way to handle guest accounts is to install them on an as-needed basis, and delete them as soon as the people using them leave. Guest accounts should never be given simple passwords such as "guest" or "visitor," and should never be allowed to remain in the password file when they are not being used (Curry,

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
If NOT running NIS, browse the /etc/passwd file to determine if there is a guest account using the following command:

```
#/usr/bin/vi /etc/passwd
```

OR if RUNNING NIS, determine if there is a guest account on the system by executing the following command:

```
/bin/niscat passwd.org_dir
```

Expected Results:
Guest accounts should not exist.

Comments:
If a Guest account is present and has been approved for use, the Guest account should not have a trivial password. Try logging into the account using simple passwords such as "guest" and "visitor".

Topic: I&A
SubTopic: Password Management

Objective 71

Ensure authentication data is protected from being accessed by unauthorized users.

Rationale:

It is no longer considered secure to place even encrypted passwords in the world-readable /etc/passwd file. As a result, numerous vendors have introduced shadow password files. These files have the same encrypted passwords, but the passwords are stored in special files that cannot be read by most users on the system (Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

3.2.1.5 The COE shall protect authentication data from being accessed by unauthorized users.

Test Actions:

Step: 1

Required Action:

```
ls -ld /etc/shadow
```

Expected Results:

The following permissions are displayed:

```
   -r-------- root sys /etc/shadow
```

Comments:

Topic: **MAIL**
SubTopic:

Objective 32
Verify the "decode" and "uudecode" aliases have been removed from the aliases file (/etc/aliases or /usr/lib/aliases).

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Type the following command:

```
#vi /etc/aliases
```

Search for decode by typing "/decode" and press return.

Expected Results:
A message should be printed to the bottom of the window as follows:

```
Pattern not found
```

OR the decode alias line appears as follows:

```
#decode:  "|/usr/bin/uudecode"
```

Comments:
After modifying the /etc/aliases file the /etc/newaliases executable must be executed.

Topic: **MAIL**
SubTopic: **Penetration Test**

Objective 33

Verify sendmail does not support the wiz command.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

Type in the following commands:

```
    % telnet localhost 25
   wiz
   quit
```

Expected Results:

Sendmail should respond to the wiz command with "5nn error return" (e.g., "500 Command unrecognized"). Any response from the server indicating recognition of the command indicates a sendmail vulnerability and sendmail should be replaced.

The session should appear similar to the following:

```
user>telnet localhost 25
Trying 127.0.0.1 ...
Connected to localhost.
Escape character is '^]'.
220 ziggy. Sendmail 5.x/SMI-SVR4 ready at Fri, 18 Oct 1996
15:48:03 -0400
wiz
500 Command unrecognized
quit
221 ziggysol24. closing connection
Connection closed by foreign host.
user>
```

Comments:

Topic: MAIL
SubTopic: Penetration Test

Objective 34

Verify sendmail does not support the debug command.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
% telnet localhost 25
debug
quit
```

Expected Results:

Sendmail should respond to the debug command with "5nn error return" (e.g., "500 Command unrecognized"). Any response from the server indicating recognition of the command indicates a sendmail vulnerability and sendmail should be replaced.

The session should appear similar to the following:

```
user>telnet localhost 25
Trying 127.0.0.1 ...
Connected to localhost.
Escape character is '^]'.
220 ziggy. Sendmail 5.x/SMI-SVR4 ready at Fri, 18 Oct 1996
15:48:03 -0400
debug
500 Command unrecognized
quit
221 ziggysol24. closing connection
Connection closed by foreign host.
user>
```

Comments:

Topic: **MAIL**
SubTopic: **Penetration Test**

Objective 35

Verify sendmail does not support the kill command.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
   % telnet localhost 25
   kill
   quit
```

Expected Results:

Sendmail should respond to the kill command with "5nn error return" (e.g., "500 Command unrecognized"). Any response from the server indicating recognition of the command indicates a sendmail vulnerability and sendmail should be replaced.

The session should appear similar to the following:

```
user>telnet localhost 25
Trying 127.0.0.1 ...
Connected to localhost.
Escape character is '^]'.
220 ziggy. Sendmail 5.x/SMI-SVR4 ready at Fri, 18 Oct 1996
15:48:03 -0400
kill
500 Command unrecognized
quit
221 ziggysol24. closing connection
Connection closed by foreign host.
user>
```

Comments:

Topic: MAIL
SubTopic: Sendmail

Objective 31
Verify sendmail is configured correctly.

Rationale:
Electronic mail is one of the main reasons for connecting to outside networks. On most versions of Berkeley-derived UNIX systems, including those from Sun, the sendmail program is used to enable the receipt and delivery of mail. Because of its design, sendmail runs as the superuser, making its security holes a significant problem for the entire system. As with the FTP software, older versions of sendmail have several bugs that allow security violations. One of these bugs was used with great success by the Internet worm (Curry, 1990).

DII COE SRS Requirement:

Test Actions:
Step: 1
Required Action:
If you use a vendor version of sendmail, ensure that you have installed the latest patches as sendmail(8) has been a source of a number of security vulnerabilities. Refer to AUSCERT Advisories SA-93:10, AA-95.08 and AA-95.09b and CERT Advisories CA-94:12, CA-95:05 and CA-95:08.

Browse the /etc/mail/sendmail.cf and verify the following lines:

```
Mlocal, P=/bin/mail, F=rlsDFMmnP, S=10, R=20, A=mail -d $u
# Mprog, P=/bin/sh, F=lsDFMeuP, S=10, R=20, A=sh -c $u
Mprog, P=/bin/true, F=lsDFMeuP, S=10, R=20, A=true
```

Expected Results:
Sendmail should be properly configured.

Comments:

Step: 2
Required Action:
Enter the following command:

```
#vi /etc/mail/sendmail.cf
```

Expected Results:
Any line starting with "OW" only has a "*" next to it (Or does not exist).

The options part of the general configuration information section includes lines similar to:

```
    # log level
    OL9
```

OR (for sendmail 8.7 or later)

```
    # log level
    O LogLevel=9
```

(The higher the number, the more information is logged)

Topic: **MAIL**
SubTopic: ***Sendmail***

The Local and Program Mailer specification section contains a commented out Mprog entry similar to the following:

```
#Mprog,  P=/bin/sh,   F=lsDFMeuP,  S=10, R=20, A=sh -c $u
```

OR a modified Mprog line similar to the following:

```
Mprog,  P=/bin/true,   F=lsDFMeuP,  S=10, R=20, A=true
```

Comments:
Sendmail doesn't deliver mail, it invokes the program listed on the Mlocal line in the sendmail.cf file (after setuiding itself to the receiving user). You'll have to check out the capabilities of that program to be sure (although sendmail 8 comes with a binmail delivery program which doesn't do any forwarding).

Step: 3

Required Action:
Type the following command:

```
#vi /etc/mail/mailx.rc
```

Expected Results:
The following lines appear as specified:

```
set append dot
if t
   set SHELL=/bin/true
else
   set SHELL=/bin/true
endif
```

Comments:

Step: 4

Required Action:
As root execute the following command:

```
#find / -name .forward -exec ls -ald {} \; -exec more {} \;
```

Expected Results:
There are no .forward files listed.

Comments:
If the responsible person permits .forward files, any .forward files in user home directories do not execute an unauthorized command or program.

Step: 5

Required Action:
Enter the following command:

```
#vi /etc/syslog.conf
```

Topic: MAIL
SubTopic: Sendmail

Expected Results:
The file syslog.conf contains lines similar to:

```
   mail.info      /dev/console
   mail.info      /var/adm/message
```

The white space between the syslog.conf entries must be a tab character.

Comments:

These lines cause mail informational messages to be written to the console and to the messages file.

Step: 6

Required Action:
Review the /etc/aliases file from an administrator command tool using the following command:

```
#/usr/bin/vi /etc/aliases
```

Expected Results:
- MAILER-DAEMON is redirected to Postmaster
- audit_warn is redirected to the system administrator's account
- nobody is redirected to /dev/null
- The decode alias is commented out or not present
- All programs executed by an alias are owned by root
- All programs executed by an alias have permissions 755
- All programs executed by an alias are stored in a root owned systems directory such as /usr/local/bin

Comments:
/etc/aliases is used to create administrative mail aliases. The mail aliases are recognized by sendmail for the local host.

Topic: MAIL
SubTopic: Sendmail bug Penetration Tests

Objective 274

Verify that the sendmail -d bug does not exist.

Rationale:

DII COE SRS Requirement:

Test Actions:

***Step:* 1**

Required Action:

From a command shell, execute the following command:

```
# /usr/lib/sendmail -d3294967296
```

Expected Results:

This command does not cause a segmentation fault.

Comments:

On some versions of sendmail it is possible to get root access by supplying greater than normal address space ranges that are used in its array index to the -d flag. If this causes a segmentation fault then you'll likely have a bug in your version of sendmail. The problem is that numbers in this range may skip the range checks and result in accessing negative indexes into the debug array. Hence it is possible to write to locations in memory before the debug array.

Topic: Markings
SubTopic:

Objective 6

Verify a security warning is displayed prior to the login process indicating restrictions that apply to logins, the highest classification of information processed on the system, and that misuse is subject to applicable penalties.

Rationale:

DII COE SRS Requirement:

3.2.7.1 The COE shall display a security warning prior to the login process that indicates the highest classification of information processed on the system and that misuse is subject to applicable penalties.

Test Actions:

Step: 1

Required Action:

Prior to login view the monitor. Review the /etc/motd file and verify that the text in the file contains the text that is the site approved warning to users logging on the system.

Expected Results:

A security warning is displayed prior to the login process indicating restrictions that apply to logins, the highest classification of information processed on the system, and that misuse is subject to applicable penalties.

Comments:

DII COE does not use /etc/motd.

Topic: **NETWORK CONFIGURATION**
SubTopic:

Objective 41
Verify the network services are appropriately configured and defined.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: ***1***

Required Action:
Verify that the network services are configured securely by browsing the /etc/inetd.conf file using the following command:

```
#/usr/bin/vi /etc/inetd.conf
```

Expected Results:
Unnecessary network services should be disabled.

A "#" starts each line identifying a disabled service. Verify that the following services are disabled:

name, shell, login, exec, comsat, talk, uucp, finger,
systat, netstat, admind, rquotad, rusersd, sprayd,
walld, rstatd, rexd, rpc.cmsd, rpc.ttdbserverd

Comments:
Services to be disabled include:
name: obsolete name server protocol
shell: allows remote user via rsh to run processes on this system
login: allows remote user via rlogin
exec: allows remote users access via rexec
comsat: real-time intrusive notification to users that mail has arrived
talk: remote chat protocol
uucp: UNIX-to-UNIX copy over TCP
finger: remote access to local user information
systat: allows remote users to view the process table
netstat: allows remote users to view the list of active network connections
admind: allows remote users to execute remote administrative activities
rquotad: provides disk quota information to NFS clients
rusersd: provides local user information
sprayd: allows remote users to send a stream of IP packets to the host and have them acknowledged
walld: allows remote users to post messages to system users
rstatd: allows remote users to view system information such as load
rexd: obsolete remote execution server with no security
rpc.cmsd: calendar manager
rpc.ttdbserverd: tool talk database server that allows object linking. MAY BE NEEDED for

DCE.

tftpd: trivial ftp server.

Topic: NETWORK CONFIGURATION
SubTopic:

Step: 2

Required Action:
Verify that the permissions of the /etc/inetd.conf file are correct using the following command:

```
#/bin/ls /etc/inetd.conf
```

Expected Results:

The permissions are set to 600 and the owner is root.

Comments:

Step: 3

Required Action:
Use the following command to verify that only required and authorized network services are registered with the portmapper. The following command determines which services are registered with the Portmapper:

```
# /user/bin/rpcinfo -p localhost
```

Expected Results:
Only appropriate services are registered with portmapper

The following services are NOT listed:

name, shell, login, exec, comsat, talk, uucp, finger,
systat, netstat, admind, rquotad, rusersd, sprayd,
walld, rstatd, rexd, rpc.cmsd, rpc.ttdbserverd

Comments:

Step: 4

Required Action:
Verify that the network services are appropriately configured by browsing the /etc/inet/services file with the following command:

```
#/usr/bin/vi /etc/inet/services
```

Expected Results:
Unnecessary network services should be disabled.

A "#" starts each line identifying a disabled service. Verify that the following services are disabled:

name, shell, login, exec, comsat, talk, uucp, finger,
systat, netstat, admind, rquotad, rusersd, sprayd,
walld, rstatd, rexd, rpc.cmsd, rpc.ttdbserverd

Topic: **NETWORK CONFIGURATION**
SubTopic:

Comments:

NOTE: /etc/services is a link to /etc/inet/services.

The services database lists the names of TCP and UDP services and their well known port numbers; it is used by programs that call network services. The Solaris installation automatically creates the services database; additional entries may be added after system installation. A "#" starts each line identifying a disabled service. Verify that the following services are disabled:

name	obsolete name server protocol
shell	allows remote user via rsh to run processes on this system
login	allows remote user via rlogin
exec	allows remote users access via rexec
comsat	real-time intrusive notification to users that mail has arrived
talk	remote chat protocol

Step: 4

Required Action:

Verify that the permissions and owner of the /etc/inet/services file are correct using the following command:

```
#/bin/ls /etc/inet/services
```

uucp	UNIX-to-UNIX copy over TCP
finger	remote access to local user information
systat	allows remote users to view the process table
netstat	allows remote users to view the list of active network connections
admind	allows remote users to execute remote administrative activities
rquotad	provides disk quota information to NFS clients
rusersd	provides local user information
sprayd	allows remote users to send a stream of IP packets to the host and have them acknowledged
walld	allows remote users to post messages to system users
rstatd	allows remote users to view system information such as load
rexd	obsolete remote execution server with no security
rpc.cmsd	calendar manager
rpc.ttdbserverd	tool talk database server that allows object linking.

Expected Results:

The permissions are set to 600 and the owner is root.

Comments:

Topic: **NETWORK CONFIGURATION**
SubTopic:

Objective 43

Verify netrc files are not used.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root execute the following command:

```
#/bin/find / -name .netrc -exec ls -ld {} \; -exec more {} \;
```

Expected Results:

Any output indicates the existence of a .netrc file on the system. The file path, permissions and contents are listed. There should NOT be any output from this command.

Comments:

The .netrc file should not exist on a secure system.

If the responsible security officer has approved the use of .netrc files for a specific purpose:
Do not store password information in .netrc files.
Set Permissions on .netrc files to disallow read and write access by group and world (i.e. 600).

Topic: **NETWORK CONFIGURATION**

SubTopic:

Objective 113

Verify Subnet addresses are appropriately configured.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Review: `/etc/netmasks`.

Expected Results:

The correct subnet definitions must be obtained from the local network administrator.

Comments:

Topic: NETWORK CONFIGURATION
SubTopic: .rhost files

Objective 115

Determine if any rhost files are used on the system.

Rationale:

The .rhosts file is similar in concept and format to the hosts.equiv file, but allows trusted access only to specific host-user combinations, rather than to hosts in general. Each user may create a .rhosts file in his home directory, and allow access to his account without a password. Most people use this mechanism to allow trusted access between accounts they have on systems owned by different organizations that do not trust each other's hosts in hosts.equiv. Unfortunately, this file presents a major security problem: While hosts.equiv is under the system administrator's control and can be managed effectively, any user may create a .rhosts file granting access to whomever he chooses, without the system administrator's knowledge (Curry, 1990).

The only secure way to manage .rhosts files is to completely disallow them on the system. The system administrator should check the system often for violations of this policy (Curry, 1990).

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following command:

```
#/bin/find / -name .rhosts -exec ls -ldb {} \;
 -exec more {} \;
```

Expected Results:

There should be no output from this command. Output means that a .rhosts file has been found. Users should not have a .rhosts file.

Comments:

Cron should be used to periodically check for, report the contents of, and remove .rhosts files.

If there is a genuine need for .rhosts files (e.g., running backups over a network unattended) and their use has been approved by responsible security officer:

the first character of any .rhosts file is not "-".

The permissions of all .rhosts files are set to 600

The owner of each .rhosts file is the account's owner
No .rhosts file contains the symbol "+" on any line

Usage of netgroups within .rhosts does not allow unintended access to this account

.rhosts files do not use '!' or '#'

Topic: **NETWORK CONFIGURATION**
SubTopic: **NFS**

Objective 76
Verify the files on the server are not world-writeable or group-writeable.

Rationale:
Because the NFS server maps root to nobody, you can protect files and directories on your server by setting their owner to root and making them not world-writeable or group-writeable.

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Browse the /etc/dfs/dfstab file using the following command:

```
#vi /etc/dfs/dfstab
```

and for each shared filesystem run the following command:

```
/bin/find filesystem \( -perm -2 -o -perm -20 \) \
-exec ls -ldg {} \;
```

Expected Results:
No files should be listed.

Comments:

Topic: **NETWORK CONFIGURATION**
SubTopic: **NFS**

Objective 77

Ensure filesystems are mounted with the nosuid option and read-only where practical. If read-only is not practical, verify system files and user home directories are not mounted.

Rationale:

In some versions of UNIX, it is possible to turn off the SUID and SGID bits on mounted filesystems by specifying the nosuid option with the mount command. If available, this option should always be specified when a filesystem is mounted unless there is an overriding reason to import SUID or SGID files from the mounted filesystem (Garfinkel and Spafford, 1992).

One of the best ways to protect sensitive files and directories is to mount them on read-only disks. It is recommended that the following directories be mounted as read-only partitions: /, /usr/bin, /bin, /etc, /lib, /usr/lib, /usr/ucb (if it exists), /usr/include, /usr/src, /usr/etc (if it exists) (Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Browse the /etc/vfstab file using the following command:

```
#vi /etc/vfstab
```

Expected Results:

The flag rw should only exist if a legitimate need exists and the flag nosuid should appear.

Comments:

Each nfs entry in the /etc/vfstab file should appear similar to the following line

```
#device    device       mount        FS          fsck    mount    mount
#to mount  to fsck      point       type         pass    at      bootoptions
Exporthost:/ExportDirPath      -  /mountpoint   nfs     -    yes  ro,bg,nosuid
```

OR if mounting the filesystem by from the command line use the following command:

```
# mount -r -o nosuid,bg serv:/usr/src /usr/src
```

Topic: NETWORK CONFIGURATION
SubTopic: NFS

Objective 117

Verify the appropriate entries are in the exports file.

Rationale:

NFS is a distributed database system that is designed to allow several hosts to share files over the network. One of the most common uses of NFS is to allow diskless workstations to be installed in offices, while keeping all disk storage in a central location. As distributed by Sun, NFS has no security features enabled. This means that any host on the Internet may access your files via NFS, regardless of whether you trust them or not (Curry, 1990).

Fortunately, there are several easy ways to make NFS more secure. The more commonly used methods are described in this section, and these can be used to make your files quite secure from unauthorized access via NFS. Secure NFS, introduced in SunOS Release 4.0, takes security one step further, using public-key encryption techniques to ensure authorized access (Curry, 1990).

The file /etc/exports is perhaps one of the most important parts of NFS configuration. This file lists which file systems are exported (made available for mounting) to other systems (Curry, 1990).

The root= keyword specifies the list of hosts that are allowed to have super-user access to the files in the named file system. The access= keyword specifies the list of hosts (separated by colons) that are allowed to mount the named file system. If no access= keyword is specified for a file system, any host anywhere on the network may mount that file system via NFS (Curry, 1990).

Obviously, this presents a major security problem, since anyone who can mount your file systems via NFS can then peruse them at his leisure. Thus, it is important that all file systems listed in exports have an access= keyword associated with them. Netgroups can also be specified (Curry, 1990).

Normally, NFS translates the super-user id to a special id called "nobody" in order to prevent a user with "root" on a remote workstation from accessing other people’s files. This is good for security, but sometimes a nuisance for system administrators, since you cannot make changes to files as "root" through NFS (Curry, 1990).

The exports file also allows you to grant super-user access to certain file systems for certain hosts by using the root= keyword. Following this keyword a colon-separated list of up to ten hosts may be specified (Curry, 1990).

Granting "root" access to a host should not be done lightly. If a host has "root" access to a file system, then the super-user on that host will have complete access to the file system, just as if you had given him “root” password on the server. Untrusted hosts should never be given “root” access to NFS file systems (Curry, 1990).

DII COE SRS Requirement:

Topic: NETWORK CONFIGURATION
SubTopic: NFS

Test Actions:

Step: 1

Required Action:
Use the following command to ensure that file systems are correctly exported:

```
/usr/bin/vi /etc/dfs/dfstab
```

This file will not exist if the computer being tested is not an NFS server.

Expected Results:
Only necessary filesystems are exported.
Only authorized hosts are given access to the exported filesystems.
All entries use fully qualified hostnames (Preferably an ip address).
Filesystems are shared using "anon=-1" to disallow accesses that are not accompanied by a user ID.
The NFS server is not self-referenced, either by name or by specification of a ‘localhost’ entry.
File systems to be exported are shared as read-only, except where specifically approved by the responsible security officer.
Only the minimum access necessary is given on the exported filesystem.
File systems to be exported are shared non-setuid.
The "root = " option should NOT be used.
Access should be granted by netgroup or host.

Comments:
Use of a network file system must be approved for use by the responsible security officer.
All NFS patches have been applied.
Ensure that you never export file systems unintentionally to the world.
Review periodically what you currently have exported.
Run fsirand for all your file systems and rerun it periodically.
Ensure that the RPC portmapper does not allow proxy requests.

Step: 2

Required Action:
Execute the following command and ensure that the owner and permissions of the dfstab file are correct:

```
#/usr/bin/ls -lg /etc/dfs/dfstab
```

Expected Results:
The file /etc/dfs/dfstab has permissions 644.

The file /etc/dfs/dfstab is owned by root.

Comments:
Use of a network file system must be approved for use by the responsible security officer.
All Sun-recommended NFS patches have been applied.
Review periodically what you currently have exported.

Run fsirand for all your file systems and rerun it periodically.
Ensure that the RPC portmapper does not allow proxy requests.

Topic: NETWORK CONFIGURATION
SubTopic: NFS

Step: 3

Required Action:

NFS port monitoring is enabled.

Expected Results:

Comments:

Topic: NETWORK CONFIGURATION
SubTopic: Penetration

Objective 89

Determine whether rusers is enabled.

Rationale:

The UNIX rusers command displays information about accounts currently active on a remote system. This may provide an attacker with account names or other information useful in mounting an attack (CERT Advisory CA-93:14).

DII COE SRS Requirement:

RUSERS

Test Actions:

Step: 1

Required Action:

Type the following command from a networked host:

```
%  rusers -a <hostname>
```

Expected Results:

If the error message "<hostname>: RPC: Program not registered," then rusers is disabled. If instead, a list of user names and login information was generated, then a rusers server is running on the host.

Comments:

rusers should not be enabled unless there is a legitimate business need.

Topic: NETWORK CONFIGURATION
SubTopic: Trusted Hosts

Objective 161
Check the /etc/hosts.equiv file to verify that the default setting of "trust all hosts" has been changed. If there are individual entries in this file, verify that all entries are appropriate.

Rationale:
One of the most convenient features of the UNIX networking software is the concept of "trusted" hosts. The software allows the specification of other hosts (and possibly users) who are to be considered trusted - remote logins and remote executions from these hosts will be permitted without requiring the user to enter a password. This is very convenient, because users do not have to type their password every time they use the network. Unfortunately, for the same reason, the concept of a trusted host is also extremely insecure (Curry, 1990).

The Internet worm made extensive use of the trusted host concept to spread itself throughout the network. Many sites that had already disallowed trusted hosts did fairly well against the worm compared with those sites that did allow trusted hosts (Curry, 1990).

The file /etc/hosts.equiv can be used by the system administrator to indicate trusted hosts. Each trusted host is listed in the file, one host per line. If a user attempts to login or execute a command remotely from one of the systems listed in hosts.equiv, and that user has an account on the local system with the same login name, access is permitted without requiring a password (Curry, 1990).

Provided adequate care is taken to allow only local hosts in the hosts.equiv file, a reasonable compromise between security and convenience can be achieved. Nonlocal hosts (including hosts at remote sites of the same organization should never be trusted. Also, if there are any machines at your organization that are installed in "public" areas you should not trust these hosts (Curry, 1990).

DII COE SRS Requirement:

Test Actions:
Step: 1
Required Action:
Execute the following command:

```
%ls -ldgb /etc/hosts.equiv; /bin/more /etc/hosts.equiv
```

Expected Results:
The following response is displayed:

```
/etc/hosts.equiv: No such file or directory.
```

Comments:
Check for the presence of /etc/hosts.equiv after each operating system or patch installation.

Step: 2
Required Action:
If the responsible security officer has approved the use of a /etc/hosts.equiv file for a specific

purpose execute the following command:

```
%ls -ldgb /etc/hosts.equiv; /bin/more /etc/hosts.equiv
```

Topic: **NETWORK CONFIGURATION**
SubTopic: ***Trusted Hosts***

Expected Results:

- The owner of /etc/hosts.equiv is root.
- The permissions of /etc/hosts.equiv are set to 600.
- The first character of /etc/hosts.equiv is not '-'.
- /etc/hosts.equiv does not contain a line with only a "+" (a plus sign).
- /etc/hosts.equiv lists only a small number of trusted hosts, and all hosts listed are within your domain or under your management.
- /etc/hosts.equiv does not include '!' or '#'.
- All hosts in /etc/hosts.equiv are specified using IP addresses to mitigate DNS spoof attacks.
- Use netgroups in /etc/hosts.equiv for easier management.

Comments:

The /etc/hosts.equiv file contains a list of trusted hosts, one per line. If a user attempts to log in remotely (using rlogin) or to remotely execute a command (using rsh) from one of the hosts listed in this file, and if that user has an account on the local system with the same login name, the system allows the user to log in without a password. The /etc/hosts.equiv file may have several entries. It should be verified that each entry is appropriate. A line of the form +@host-group makes all of the hosts in the network group hostgroup trusted; likewise, a line which has the form -@anotherhostgroup makes all of the hosts in the networkgroup anotherhostgroup specifically not trusted. The file is scanned from the beginning to the end; the scanning stops after the first match. A single line of + in the hosts.equiv file indicates that every known host is trusted. This can create a serious security problem. It is recommended that the /etc/hosts.equiv file be removed

Topic: NETWORK CONFIGURATION
SubTopic: promiscuous ethernet interface

Objective 280

Verify that no interface is in promiscuous mode.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

An Ethernet interface that is running in promiscuous mode can be identified with the following command:

```
/usr/sbin/ifconfig -a | grep -i promisc
```

Expected Results:

Any output is an indication of an ethernet interface in promiscuous mode. This is usually a bad sign and the system should be examined closely to determine if ethernet sniffers are being run on the system.

Comments:

An interface in promiscuous mode will allow programs to read passwords and other data (from the network) that should be kept secret.

Topic: NFS
SubTopic: Penetration

Objective 287

Verify that a variety of known NFS bugs are not present in the system being tested.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Expected Results:

Comments:

I just read a post in comp.security.unix entitled "widespread security hole in exporting of filesystems" which claims there are ways to break into a system that has filesystems exported to itself. This hole has been known for quite a while. You can test it by writing a program, I don't think there is any way to use a normal system utility to check for the hole. To exploit call the mountproc_mnt_1() RPC only use the pmap_rmtcall() routine to call it rather than calling it through a normal clnt_call(). If your mountd is smart enough to turn down requests on non-privileged ports then you will not be vulnerable to this as the portmapper always makes requests on a non-priveledged port.

People might want to use the nfsbug detector by Leendert van Doorn. Idon't know if it's in the PD, but it will test an NFS server for several(known) security holes.

Topic: **OBJECT REUSE**
SubTopic:

Objective 13

Verify object reuse provisions are enforced by the operating system and/or by features in the application software.

Rationale:

DII COE SRS Requirement:

3.2.9.1 No information, including encrypted representations of information, produced by a prior subject's actions shall be available to any subject that obtains access to an object that has been released back to the COE.
3.2.9.2 All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation, or reallocation to a subject from the COE's pool of unused storage objects.

Test Actions:

Step: 1

Required Action:

Review Solaris SHIELD Basic Security Manual, Chapter 5; turn to the appropriate section(s) which demonstrate the ability of the NMS to satisfy the "Orange Book" requirements.

Expected Results:

Section(s) are present in the manual which verify that the component Operating System was designed to meet the C2 requirements of the "Orange Book."

Comments:

The TCSEC's object reuse requirement for computing systems at C2 level and above is fulfilled by the device allocation mechanism. The device allocation mechanism makes it possible to assign certain devices to one user at a time, so that the device can be accessed by only that user while it is assigned to that user's name.

Step: 2

Required Action:

Review the following file:

```
/etc/security/device_allocate
```

Expected Results:

The file /etc/security/device_allocate is configured so that the tape drive, floppy, CD-ROM, and audio devices are purged whenever they are allocated.

All multiuser devices should be configured as allocatable. The following entries should appear in the device_allocate file for the tape drive, floppy, CD-ROM, and audio, respectively:

```
st0;st;;;;/etc/security/lib/st_clean
fd0;fd;;;;/etc/security/lib/fd_clean
sr0;sr;;;;/etc/security/lib/sr_clean
audio;audio;;;;/etc/security/lib/audio_clean
```

Each entry should have a device clean entry.

Topic: **OBJECT REUSE**
SubTopic:

Comments:

An entry in the device_allocate file does not mean the device is allocatable, unless the entry specifically states the device is allocatable. An asterisk in the fifth field indicates to the system that the device is not allocatable, that is, the system administrator does not require a user to allocate the device before it is used nor to deallocate it afterwards.

The device clean scripts address the security requirements that all usable data is purged from a physical device before reuse. By default, cartridge tape drives, floppy disk drives, CD-ROM devices, and audio devices require device clean scripts, which are provided.

Device allocation satisfies part of the object reuse requirement. The device clean scripts make sure that data left on a device by one user is cleared before the device is allocatable by another user.
(SunSHIELD Basic Security Module Guide)

Topic: **OBJECT REUSE**
SubTopic:

Objective 122
Verify that the keyboard, mouse, console, and audio device files are owned by the user logged in.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Browse the /etc/logindevperm file using the following command:

```
#/usr/bin/vi /etc/logindevperm
```

Expected Results:
The file /etc/logindevperm contains the lines:

```
/dev/console   0600     /dev/mouse:/dev/kbd
/dev/console   0600     /dev/sound/*          # audio devices
/dev/console   0600     /dev/fbs/*            # frame buffers
```

The file /etc/logindevperm is owned by root and has permissions 644.

Read the man page for logindevperm(4) for more information.

Comments:
Solaris versions 2.3 and above have a protection facility for framebuffers which is a superset of the functionality provided by /etc/fbtab in SunOS 4.1.x.

Under Solaris, /dev/fbs is a directory that contains links to the framebuffer devices. The /etc/logindevperm file contains information that is used by login(1) and ttymon(1M) to change the owner, group, and permissions of devices upon logging into or out of a console device. By default, this file contains lines for the keyboard, mouse, audio, and frame buffer devices.

Topic: SYSTEM ARCHITECTURE
SubTopic:

Objective 142

Determine if any development tools exist on the system.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

To verify that development tools such as language compilers, linkers, and debuggers can be accessed only by authorized users.

Expected Results:

Unprivileged users cannot access the development tools listed below:

/usr/bin/adb
/usr/bin/as
/usr/bin/bc
/usr/lib/compile
/usr/bin/cb
/usr/bin/cflow
/usr/bin/cxref
/usr/bin/dbxtool
/usr/bin/ld
/usr/bin/lex
/usr/bin/m4
/usr/bin/od
/usr/bin/rpcgen
/usr/bin/yacc
/usr/bin/dbx
/usr/bin/gcore
/usr/bin/sccs
/usr/bin/xstr
/usr/openwin/bin/cps
/usr/openwin/bin/makeafb
/usr/5lib/compile
/usr/5bin/lint
/usr/5bin/od

Comments:

Step: 2

Required Action:

Verify that the development tools listed above are owned by a privileged user and cannot be accessed by an unprivileged user.

For each of the development tools listed above, enter "`ls -alg <development tool>`".

Topic: **SYSTEM ARCHITECTURE**
SubTopic:

Objective 143

Verify development tools such as language compilers, linkers, and debuggers are adequately protected and can only be accessed by authorized users.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Verify that the development tools listed in the comments are owned by a privileged user and cannot be accessed by an unprivileged user. For each of the development tools listed, enter "`ls -alg <development tool>`".

```
find / -name gcc -exec ls -ld {}
```

Expected Results:

The permissions on the tool executables should be 750. The development tools should be assigned to a specific developer's group.

Comments:

For operational systems, development tools such as language compilers, linkers, and debuggers are not available on the system. If the responsible security officer has approved the use of specific development tools such as language compilers, linkers, and debuggers on an operational system for a specific purpose, the development tools can be accessed only by authorized users. The following development tools should not be accessible to unprivileged users:

/usr/bin/adb
/usr/bin/as
/usr/bin/bc
/usr/lib/compile
/usr/bin/cb
/usr/bin/cflow
/usr/bin/cxref
/usr/bin/dbxtool
/usr/bin/ld
/usr/bin/lex
/usr/bin/m4
/usr/bin/od
/usr/bin/rpcgen
/usr/bin/yacc/usr/ucb/dbx
/usr/ucb/gcore
/usr/ucb/sccs
/usr/ucb/xstr
/usr/openwin/bin/cps
/usr/openwin/bin/makeafb

```
/usr/5lib/compile
/usr/5bin/lint
/usr/5bin/od
```

Topic: SYSTEM ARCHITECTURE
SubTopic:

Objective 148
Verify the installation defaults file is configured correctly.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Execute the following command:

```
$/usr/ucb/vi /var/sadm/install/admin/default
```

Expected Results:
The following parameters should be set:

- The mail parameter either should not be present or a system administrative account should be specified.
- The runlevel parameter should be set to quit or ask.
- The conflict parameter either should be set to quit or ask.
- The setuid parameter should NOT be set to nocheck or ask.
- The action parameter should be set to quit or ask.

Comments:
Solaris 2.4 system software is delivered in units known as packages. A package is a collection of files and directories required for a software product.

admin is a generic name for an ASCII file that defines default installation actions by assigning values to installation parameters. For example, it allows administrators to define how to proceed when the package being installed already exists on the system.

The default admin file is located in /var/sadm/install/admin/default. If the -a option is not used when installing a package with the -a option of pkgadd, the default admin file is used.

The following parameters are among those that may be specified:

- mail - Defines a list of users to whom mail should be sent following installation of a package. If the list is empty, no mail is sent. If the parameter is not present in the admin file, the default value of root is used.

- runlevel - Indicates resolution if the run level is not correct for the installation or removal of a package.

Options are nocheck, which does not make a check for run level, and quit, which aborts installation if the run level is not met.

- conflict - Specifies what to do if an installation expects to overwrite a previously installed file, thus creating a conflict between packages. Options are nocheck, which does not check for conflict, and quit, which aborts installation if conflict is detected.

- setuid - Checks for executables which will have setuid or setgid bits enabled after installation. Options are nocheck, which does not check for setuid executables, quit, which aborts installation if setuid processes are detected, and nochange, which overrides installation of setuid processes.

- action - Determines if action scripts provided by package developers contain possible security impact. Options are nocheck, which ignores security impact of action scripts, and quit, which aborts installation if action scripts may have a negative security impact.

Topic: SYSTEM ARCHITECTURE
SubTopic:

Objective 155

Verify the processes dispatched by the init process are appropriate. The default processes launched by the init process are: ap, fs, is, p3, s0, s1, s2, s3, s5, s6, fw, of, rb, sc, and co. Others must be explicitly approved.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Review the file:

```
/etc/inittab
```

Expected Results:

The default processes launched by the init process are: ap, fs, is, p3, s0, s1, s2, s3, s5, s6, fw, of, rb, sc, and co.

Comments:

The file /etc/inittab dispatches only executables that are the original, non-trojaned, executables. Any other processes should be explicitly approved by the DII COE chief engineer.

The file /etc/inittab controls process dispatching by the init process. The processes most typically dispatched by init are daemons. The inittab file is composed of entries that are position dependent and have the following format:

```
    id:state:action:process
```

Topic: **SYSTEM ARCHITECTURE**
SubTopic:

Objective 158

Verify the user environment is configured properly. By default, the /etc/profile file sets the user terminal type, checks for new email, and sets the umask. Any other activity should be explicitly approved.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root execute the following shell script for printing the umask value for each user:

```
#!/bin/sh
date
uname -a
PATH=/bin:/usr/bin:/usr/etc:/usr/ucb

HOMEDIRS=`cat /etc/passwd | awk -F: 'length($6)>0 {print $6}' |
sort -u`
FILES=".cshrc .login .profile "
for dir in $HOMEDIRS
do
         echo "------------------------------------------"
         echo Home Directory being checked is $dir
      for file in $FILES
      do
         ls -ald $dir/$file
         if [ -f $dir/$file ]
         then
            grep -s umask /dev/null $dir/$file
         fi
      done
done
echo "------------------------------------------"
```

Expected Results:

The umask value for each user is set to something sensible like 027 or 077.

Comments:

SCRIPT DOES NOT WORK UNDER NIS or NIS+
When a file or directory is created, it has a default set of permissions. These default permissions are determined by the value of umask in the system file /etc/profile, or in a user's .cshrc or .login file. By default, the system sets the permissions on a text file to 666, granting read and write permission to user, group, and others, and to 777 on a directory or executable. The value assigned by umask is subtracted from the default. This has the effect of denying permissions in the same way that chmod grants them. If possible, a .cshrc, .login, and .profile should be created

for each user owned by root and readable by the user with correct environment settings.

Topic: SYSTEM ARCHITECTURE
SubTopic:

Step: 2

Required Action:
Utilize the following shell script for viewing the account initialization files for each user:

```
#!/bin/sh
date
uname -a
PATH=/bin:/usr/bin:/usr/etc:/usr/ucb

HOMEDIRS=`cat /etc/passwd | awk -F: 'length($6)>0 {print $6}' |
sort -u`
FILES=".cshrc .login .profile .logout .mwmrc .Xsession .Xdefaults
.exrc .forward .rhosts"
for dir in $HOMEDIRS
do
         echo "------------------------------------------"
         echo Home Directory being checked is $dir
      for file in $FILES
      do
         ls -ald $dir/$file
         if [ -f $dir/$file ]
         then
            more $dir/$file
         fia
      done
done
echo "------------------------------------------"
```

Expected Results:
All account initialization files in user $HOME, and the default files that are used if these files are not present, have been reviewed to ensure that only acceptable actions are taken. Acceptable actions include: set user terminal type, check for new e-mail, and set a proper umask (027 or 077).   Any other actions should be explicitly approved by the responsible security officer. All user account initialization files are owned by the use (or root) and have permission 640.

Comments:
SCRIPT DOES NOT WORK UNDER NIS or NIS+. [Acceptable actions for .mwmrc and .Xsession TBD.]If possible, a .cshrc, .login, and .profile should be created for each user owned by root and readable by the user with correct environment settings.

Step: 3

Required Action:
Ensure the default account initialization files are secure. A shell script for viewing the files follows:

```
#!/bin/sh
date
#!/bin/sh
date
```

```
uname -a
echo ---------------------------------------------
echo /etc/.profile
echo ---------------------------------------------
ls -al /etc/.profile
cat /etc/.profile
echo ---------------------------------------------
echo /etc/skel/local.cshrc
echo ---------------------------------------------
ls -al /etc/skel/local.cshrc
cat /etc/skel/local.cshrc

echo ---------------------------------------------
echo /etc/skel/local.login
echo ---------------------------------------------
ls -al /etc/skel/local.login
cat /etc/skel/local.login

echo ---------------------------------------------
echo /etc/skel/local.profile
echo ---------------------------------------------
ls -al /etc/skel/local.profile
cat /etc/skel/local.profile

echo ---------------------------------------------
echo /etc/profile
echo ---------------------------------------------
ls -al /etc/profile
cat /etc/profile

echo ---------------------------------------------
echo ---------------------------------------------
echo DII COE initialization files
echo ---------------------------------------------
echo ---------------------------------------------
echo /etc/csh.login
echo ---------------------------------------------
ls -al /etc/csh.login
cat /etc/csh.login

echo ---------------------------------------------
echo /etc/dt/config/sys.dtprofile
echo ---------------------------------------------
ls -al /etc/dt/config/sys.dtprofile
cat /etc/dt/config/sys.dtprofile

echo ---------------------------------------------
echo /h/USERS/local/sysadmin/Scripts/.cshrc
echo ---------------------------------------------
ls -al /h/USERS/local/sysadmin/Scripts/.cshrc
cat  /h/USERS/local/sysadmin/Scripts/.cshrc
```

```
echo ---------------------------------------------
echo /h/USERS/local/sysadmin/Scripts/.login
echo ---------------------------------------------
ls -al /h/USERS/local/sysadmin/Scripts/.login
cat  /h/USERS/local/sysadmin/Scripts/.login

echo ---------------------------------------------
echo /h/COE/Scripts/.cshrc.COE
echo ---------------------------------------------
ls -al /h/COE/Scripts/.cshrc.COE
cat /h/COE/Scripts/.cshrc.COE
echo ---------------------------------------------
echo /h/COE/Scripts/.login.COE
echo ---------------------------------------------
ls -al /h/COE/Scripts/.login.COE
cat /h/COE/Scripts/.login.COE

echo ---------------------------------------------
echo /h/COE/Scripts/.xsession.COE
echo ---------------------------------------------
ls -al /h/COE/Scripts/.xsession.COE
cat /h/COE/Scripts/.xsession.COE

echo ---------------------------------------------
echo    $COE_HOME/Scripts
echo ---------------------------------------------
ls -alg  $COE_HOME/Scripts
echo ---------------------------------------------
```

Expected Results:

All default account initialization files that are used if user account initialization files are not present have been reviewed to ensure that only acceptable actions are taken. Acceptable actions include: set user terminal type, check for new e-mail, and set a proper umask (027 or 077).   Any other actions should be explicitly approved by the responsible security officer.

The default account initialization files are owned by root and have permissions 644.

Comments:

/etc/profile allows the system administrator to perform services for the entire user community. The file $HOME/.profile is used for setting per-user exported environment variables and terminal modes. Care must be taken in providing system-wide services in /etc/profile.

Step: ***4***

Required Action:

As root, execute the following command:

```
/bin/find / -name ".exrc"  -print -exec ls -ld {} \; \
-exec /usr/bin/more {} \;
```

Topic: **SYSTEM ARCHITECTURE**
SubTopic:

Expected Results:
There are no .exrc files on the system or the "exrc" option for each user is set to "noexrc".

Comments:
The editing environment defaults to certain configuration options. When an editing session is initiated, vi attempts to read the EXINIT environment variable. If it exists, the editor uses the values defined in EXINIT, otherwise the values set in $HOME/.exrc are used. If $HOME/.exrc does not exist, the default values are used.

To use a copy of .exrc located in the current directory other than $HOME, set the exrc option in EXINIT or $HOME/.exrc. Options set in EXINIT can be turned off in a local .exrc only if exrc is set in EXINIT or $HOME/.exrc.

Topic: SYSTEM ARCHITECTURE
SubTopic:

Objective 159
Verify the window tool scripts are appropriately configured.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Browse the following file:

```
/usr/openwin/lib/openwin-init
```

Expected Results:
The filemgr and postmaster should not be executed.

Comments:
Contains scripts that run when executing a window tool. The scripts can be modified. /usr/openwin/lib contains configuration files for the window system. openwin-init holds OpenWindows default initialization information.

Step: 2

Required Action:
Browse the /usr/openwin/lib/openwin-menu file.

Expected Results:
Unnecessary menu items should be commented out.

Comments:
openwin-menu holds the default OpenWindows root menu.

Step: 3

Required Action:
Browse the /usr/openwin/lib/openwin-sys file.

Expected Results:
The autolockscreen should be appropriately configured. Unnecessary settings should be commented out.

Comments:
openwin-sys holds the OpenWindows system initialization information.

Topic: **SYSTEM ARCHITECTURE**
SubTopic:

Objective 188

Verify security support tools are provided to periodically determine the security posture of systems, to validate the strength of the authentication mechanism, and to determine changes to designated systems and application files.

Rationale:

DII COE SRS Requirement:

3.2.15.6 The COE shall provide the SGSO a standard set of security support tools to periodically determine the security posture of COE systems.
3.2.15.6.1 The COE shall provide the capability to validate the strength of the authentication mechanism. For example, the capability will check for potentially weak passwords.
3.2.15.6.2 The COE shall provide the capability to determine changes to designated systems and applications files, e.g., password or rc.* files.

Test Actions:

Step: 1

Required Action:

Review the file:

```
/usr/aset/asetenv
```

Expected Results:

The ASET should be scheduled to run on a regular basis, and should check all system files and any other security-relevant files added to the system. The cron entry should look something like the following example:

```
   0 0 * * * /usr/aset/aset  -l med -d /usr/aset
```

The ASET should check all system files and any other security-relevant files added to the system. Look for an ASET entry in root’s cron jobs. ASET should be configured to tune the system. The /usr/aset/masters/cklist.med file is correct. The file /usr/aset/asetenv is set so the ASET checks all system files and any other security-relevant files added to the system. The file /usr/aset/asetenv is set so the ASET checks system files, users and groups, system configuration, environment, and eeprom. The root crontab file contains an ASET entry that runs ASET regularly (preferably daily) and checks security at least at the medium security level. Baseline alterations are audited alterations. Security administrator should check that any ASET-discovered security weaknesses are corrected, if possible.

Comments:

ASET should be scheduled to run regularly (preferably daily) and to check security at least at the medium security level.

ASET depends on a correctly established and maintained configuration baseline for the kernel. The correct functioning of ASET requires the security administrator to check that proper kernel baseline updates are made. The auditing of all baseline alterations will notify the system administrator of any improper alterations. At the level ASET has to run in DII COE version 3.0, ASET performs a number of security checks. The security administrator should check that any ASET-discovered security weaknesses are corrected, if possible.

Topic: SYSTEM ARCHITECTURE
SubTopic: Loaded OS Modules

Objective 149

Determine the OS modules that have been installed on the system.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

From the command line as root type:

```
#modinfo
```

Expected Results:

The specific modules that are approved is hardware-dependent.
Only approved kernel modules are present in the directory that contains the dynamically loadable kernel modules. The directory is specified by the "moddir" variable, set in the file /etc/system).

Comments:

The modinfo command displays information about the loaded modules. The format of the information is as follows:

```
Id  Loadaddr  Size  Info  Rev  Module Name
```

where Id is the module ID, Loadaddr is the starting text address, size is the size of text, data, and bss in bytes, Info is module specific info, Rev is the revision of the loadable modules system, and Module Name is the filename and description of the module.

Topic: SYSTEM ARCHITECTURE
SubTopic: Operating System

Objective 151

Verify the system kernel configuration file is correct.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Review the file:

/etc/system

Expected Results:

The system kernel configuration file /etc/system contains:

```
# Enable C2 Audit
set c2audit:audit_load = 1
# Enable NFS port monitoring
set nfs:nfs_portmon = 1
```

For DII COE the following additional settings should be present:

```
set shmsys:shminfo_shmmax=0x4000000
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=128
set enable_sm_wa=1
```

Comments:

The system file is used for customizing the operation of the operating system kernel. The recommended procedure is to preserve the original system file before modifying it.

The boot program contains a list of default kernel modules to be loaded. The /etc/system configuration file, which is read at boot time, can be used to override the list of default modules. Care should be used when modifying the system file; it modifies the operation of the kernel.

Topic: SYSTEM ARCHITECTURE
SubTopic: Operating System

Objective 153

Verify the appropriate operating system patches have been applied.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

From the command line as root type:

```
#showrev -a
```

Expected Results:

As of March, 1996, the recommended patches are:

- 101945-36 jumbo patch for kernel
- 101959-06 lp jumbo patch
- 102044-01 bug in mouse code makes "break root" attack possible
- 102070-01 Bugfix for rpcbind/portmapper
- 102165-02 nss_dns.so.1 fixes
- 102216-05 klmmod and rpcmod jumbo patch
- 102218-03 libbsm fixes
- 102680-03 fixes for ufsdump and wall
- 102711-01 Creation of /tmp/ps_data is security problem
- 102922-03 inetd fixes
- 102664-01 data fault in scanc() due to bad "cp" argument
- 102292-02 OpenWindows 3.4: filemgr (ff.core) fixes
- 102756-01 Expreserver patch (This patch is not a "Sun recommended" patch).

Comments:

showrev displays revision information for the current hardware and software. With no arguments, showrev shows the system revision information including hostname, hostid, release, kernel architecture, application architecture, hardware provider, domain, and kernel version. The -a option prints all system revision information available. Window system and patch information are added.

Current operating system patch recommendations can be obtained from the SunSolve software or from the following FTP site:

sunsite.unc.edu/pub/sun-info/sun-patches/Solaris2.4.Patches

Some patches may re-enable default configurations. For this reason, it is important to go through this checklist after installing any new patches or packages.

Verify the digital signature of any signed files. Tools like PGP may be used to sign files and to verify those signatures. If an md5(1) checksum is supplied, then verify the checksum information to confirm that a valid copy has been retrieved. If only a generic sum(1) checksum is provided, then check that.

Topic: SYSTEM ARCHITECTURE
SubTopic: Printer Definition

Objective 154
Verify only appropriate printers are defined.

Rationale:

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Browse the /etc/lp/printers directory using the following command:

```
ls -ls /etc/lp/printers
```

Expected Results:
Only appropriate local and remote printers should be defined.

Comments:
This directory contains queues and configuration files for various printers and is set up by admintool. One configuration file is users.deny that denies specified users from using a particular printer.

Topic: System Architecture
SubTopic: Permissions

Objective 281

Verify that the crash program permissions are set correctly.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Attempt to execute the crash program as an unprivileged user by typing the following command:

```
/usr/sbin/crash
```

Expected Results:

An error similar to the following is produced:

```
/usr/sbin/crash:  Permission denied
```

Comments:

crash(1) allows you to snoop through kmem too (inherited from Solaris)

Topic: System Architecture
SubTopic: Telnet bug

Objective 278

Verify that the telnet bug does not exist.

Rationale:

There is a security hole in some versions of telnet that will allow any user on the system to overwrite any file. Using the command will overwrite any file in any filesystem with a zero-length root-owned file.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following commands:

```
#/usr/ucb/vi /tmp/file1
```

Insert some text, save the file and exit the editor. Type the following commands:

```
#ls /tmp/file1
#/usr/ucb/more /tmp/file1
```

Expected Results:

The file size of /tmp/file1 is larger than 0 and the text inserted into file1 is displayed on the screen.

Comments:

Step: 2

Required Action:

As an unprivileged user, execute the following command:

```
$/usr/bin/telnet -n /tmp/file1localhost
$ls /tmp/file1
```

Expected Results:

The file size is NOT 0.

Comments:

If the file size of /tmp/file1 is 0, the telnet daemon must be replaced.

Topic: **SYSTEM ARCHITECTURE**
SubTopic: **Operating System**

Objective 152

Determine the OS version installed. Verify that it is the correct version.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type in the following command:

```
#uname -a
```

Expected Results:

Output similar to the following is printed to the screen:

```
SunOS ziggysol24 5.4 generic sun4m sparc
```

Comments:

The most important parts are the "SunOS" and the "5.4" portions that indicate that the host being tested is running the Solaris 2.4 operating system.

Topic: **TELNET**
SubTopic:

Objective 160

Verify a user is always prompted for a password when telneting into the host machine.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

Logon to a test account. Attempt to telnet typing the command "telnet localhost". The system should respond with the login prompt. Enter a valid username.

Expected Results:

Should be prompted for a password.

Comments:

Topic: UUCP
SubTopic: Penetrate

Objective 172

Verify known UUCP bugs have been fixed.

Rationale:

UUCP is one of the oldest major subsystems of UNIX, and has had its share of security holes. All of the known security problems have been fixed in recent years. Unfortunately, there are still many old versions of UUCP in use.

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Perform the following tests of UUCP:

The mail system should not allow mail to be sent directly to a file. Test whether the system allows mail to be sent to a file with the command sequence:

```
$ mail /tmp/mailbug
 this is a mailbug file test
 ^D
```

Expected Results:

If the file mailbug appears in the /tmp directory, then the mailer is unsecure.

Comments:

Remove and replace the uucp software.

Step: 2

Required Action:

As a non privileged user, execute the following command sequences:

```
$ uux - mail ‘root `/bin/touch /tmp/foo`’
this is a mailbug command test
^D
$ uux - mail ‘root & /bin/touch /tmp/foo’
this is another test
^D
```

Expected Results:

Mail should be returned saying that `/bin/touch /tmp/foo` is an unknown user. If the mailer executed the touch (a foo file will be created in the /tmp directory) then the uux program is unsecure.

Comments:

The UUCP system should not allow a command to be encapsulated in addresses to prevent system execution of commands encapsulated in addresses.

Topic: UUCP
SubTopic: Penetrate

Step: 3

Required Action:
As a non privileged user, execute the following command sequences:

```
   $ uux - mail ‘root & /bin/touch /tmp/foo`’
   this is another mailbug command test
   ^D
   $ uux - mail ‘root & /bin/touch /tmp/foo’
   this is another test
   ^D
```

Expected Results:
Mail should be returned saying that `/bin/touch /tmp/foo` is an unknown user. If the mailer executed the touch (a foo file will be created in the /tmp directory) then the uux program is unsecure.

Comments:
The UUCP system should not allow a command to be encapsulated in addresses to prevent system execution of commands encapsulated in addresses.

Topic: UUCP
SubTopic: Disabled

Objective 288
Verify that uucp is not enabled.

Rationale:
UUCP is one of the oldest major subsystems of UNIX, and has had its share of security holes. All of the known security problems have been fixed in recent years. Unfortunately, there are still many old versions of UUCP in use.

DII COE SRS Requirement:

Test Actions:
Step: 1

Required Action:
Use the following commands to ensure that uucp is not enable or installed on the system:

```
#/usr/ucb/vi   /etc/inetd.conf
```

Expected Results:
The uucp entry in /etc/inetd.conf should NOT be enabled (i.e., the first character on the line for uucp should be
a "#").

Comments:
UUCP is one of the oldest major subsystems of UNIX, and has had its share of security holes. Although the design is not secure, the known security holes have been fixed in recent years. Unfortunately, there are still many old versions of UUCP in use.

Step: 2

Required Action:
As root, execute the following command to ensure that uucp is not installed on the system:

```
/bin/find / \( -user uucp -o -name "continuum" \) \
-exec ls -ldb {} \;
```

Expected Results:
There should be no output from this command. These daemons handle UUCP file transfers and command executions and should not exist.

Comments:
If uucp is being used, verify that the UUCP programs are owned by uucp and not by root and have the proper permissions by executing the command below as root:

```
/bin/find / \( -name uuxqt -o -name uucico -o -name \
uusched -o -name in.uucpd -o -name uux -o -name uucp \) \
-exec  ls -ldb {} \;
```

The uucp programs should run SUID uucp, not SUID root. Other than being able to read the spooled UUCP files, The uucp user doesn’t have any special privileges. The output should appear similar to the output below:

Topic: UUCP
SubTopic: Disabled

```
# /bin/find / \( -name uuxqt -o -name uucico -o -name uusched -o -name
in.uucpd
-o -name uux -o -name uucp \) -exec  ls -ldb {} \;
---s--x--x 1 uucp uucp       64240 Jul 15  1994 /usr/bin/uucp
---s--x--x 1 uucp uucp       68040 Jul 15  1994 /usr/bin/uux
drwxr-xr-x 2 uucp uucp         512 Aug 20 16:47 /usr/lib/uucp
---s--x--x 1 uucp uucp      169096 Jul 15  1994 /usr/lib/uucp/uucico
---s--x--x 1 uucp uucp       32016 Jul 15  1994 /usr/lib/uucp/uusched
---s--x--x 1 uucp uucp       81040 Jul 15  1994 /usr/lib/uucp/uuxqt
-r-xr-xr-x 1 uucp uucp        8320 Jul 15  1994 /usr/sbin/in.uucpd
-rw-rw---- 1 uucp mail         376 Oct 14 23:45 /var/mail/uucp
-r--r--r-- 1 root sys          215 Aug 20 16:47 /var/spool/cron/crontabs/uucp
```

***Step:* 3**

Required Action:

Verify that the Permissions file is properly configured using the following command:

```
#/usr/bin/vi   /etc/uucp/Permissions
```

Expected Results:

If the uucp entry is enabled, the /etc/uucp/Permissions file should allow minimal access (an empty Permissions file provides minimal access).

Comments:

The /etc/uucp/Permissions file specifies the permissions that remote computers have with respect to login, file access, and command execution. There are options that restrict the remote computer's ability to request files and its ability to receive files queued by the local machine. Another option is available that specifies the commands that a remote machine can execute on the local computer.

There are two types of Permissions file entries:

- LOGNAME Specifies the permissions that take effect when a remote computer logs into (calls) the local
computer.

- MACHINE Specifies permissions that take effect when the local computer logs into (calls) a remote host.

When using the Permissions file to restrict the level of access granted to remote computers, the following
issues should be considered:

- All login IDs used by remote computers to log in for UUCP communications must appear in one LOGNAME
entry.

- Any site that is called whose name does not appear in a MACHINE entry, will have the following default permissions or restrictions:

Topic: UUCP
SubTopic: Disabled

- Local send and receive requests will be executed.
- The remote computer can send files to the local computer's /var/spool/uucppublic directory.
- The commands sent by the remote computer for execution on the local computer must be one of the default

commands, usually rmail.

REQUEST Option When a remote computer calls the local computer and requests a file, this request can be granted or denied. The REQUEST options specifies whether the remote computer can request to set up file transfers from the local computer. The default value is REQUEST=no.

READ and WRITE Options These options specify the various parts of the file system that uucico can read from or write to. The default for both the READ and WRITE options is the uucppublic directory, /var/spool/uucppublic.

COMMANDS Option. The COMMANDS option in MACHINE entries can specify the commands that a remote computer can execute on the local computer. The COMMANDS option should be used with great care as with excessive privileges.

Step: 4

Required Action:
Verify any UUCP jobs entered in crontab are run as the user uucp and the script file is owned by root.

Expected Results:
Jobs are run as user uucp and script files are owned by root.

Comments:
crontab should run all uucp scripts as the user uucp, rather than as the user root to prevent jobs from running However, the scripts themselves should be owned by root, not uucp, so they can't misuse can compromise the security of a computer.be modified by people using the uucp programs.

Step: 5

Required Action:
Determine if the system has enabled UUCP callback.

Expected Results:
UUCP callback is enabled if possible.

Comments:
Version 2 UUCP has a callback feature that can be used to enhance security. With callback, when a remote system calls the local computer, the system immediately hangs up on the remote system and calls back. No special callback hardware is required to take advantage of UUCP callback, because it is performed by the system software, not by the modem. Note that only one system out of each pair of communicating systems can have callback enabled.

Topic: UUCP
SubTopic: ***Disabled***

Step: ***6***
Required Action:
Verify uucp's home directory is in an appropriate directory using the following commands:

```
$grep uucp /etc/passwd
$ls -ld  `grep uucp /etc/passwd | awk -F: 'length($6)>0 {print
$6}'`
```

Expected Results:
The uucp home directory should not be in a directory that is world writeable.

Comments:
The home directory for the uucp account should not be in the directory /usr/spool/uucp/uucppublic, or any other directory that can be written to by a uucp user.

Step: ***7***
Required Action:
Use the following command to ensure that there is no .rhosts file in the uucp home directory:

```
#find  `grep uucp /etc/passwd | awk -F: 'length($6)>0 {print
$6}'`  -name .rhosts -exec ls -ldb {} \;
```

Ensure that no uucp owned files or directories are world writeable.

Expected Results:
There should be no output from this command.
Comments:

Step: ***8***
Required Action:
As root ensure that no uucp owned files or directories are world writeable using the following command:

```
find / -user uucp -perm -2 -exec ls -ldb {} \;
```

Expected Results:
There is no output indicating no files on the system that are owned by uucp and world writeable.

Comments:

Topic: **WWW-HTTPD**
SubTopic:

Objective 175

Verify the http server daemon is not being run as root, but as a specially created nonprivileged user such as httpd.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following command:

```
#/bin/find / -name "*http*" -exec ls -ldb {} \;
```

Expected Results:

File permission listing reveals that the owner of the http server daemon (usually httpd) is not root and not SUID, but as a specially created nonprivileged user such as "httpd."

Comments:

Topic: **WWW-HTTPD**
SubTopic:

Objective 176

Verify httpd client processes are not being run as root.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: ***1***

Required Action:

Use the following command to verify that the http client applications are not being run as root:

```
#/usr/bin/find / -name "*osaic*" -exec ls -ldb {} \;
#/usr/bin/find / -name "*etscape*" -exec ls -ldb {} \;
```

Expected Results:

The file permissions on all http clients listed are not owned by root and are not SUID.

Comments:

Topic: **X WINDOW SYSTEM**
SubTopic: **Use of xauth access control**

Objective 183

Verify the system uses the xauth X server access control mechanism instead of the xhosts mechanism.

Rationale:

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following command:

```
/bin/find /etc -name "*rc*" -type f  -exec ls -lgdb {} \; -exec
/bin/grep xdm  {} \;
```

Expected Results:

xdm is initiated with -auth $HOME/.Xauthority.

Comments:

Step: 2

Required Action:

As an unprivileged user, execute the following command:

```
echo $XAUTHORITY
```

Expected Results:

This variable should exist and contains the magic cookie used to authenticate valid users attempting to connect to the X server. If xauth is being used and this variable is not present, then the $HOME/.Xauthority file contains the magic cookie (this is not as secure).

Comments:

Step: 3

Required Action:

As root, execute the following command:

```
/bin/find / -name xdm-config -exec ls -lgdb {} \; -exec
/usr/ucb/more {} \;
```

Expected Results:

The following lines are included:

```
DisplayManager*authorize:    true
DisplayManager*authname:   XDM-AUTHORIZATION-1
```

Comments:

The first line turns on authorization for all X servers controlled by a given xdm program. The second line sets the authority scheme to XDM-AUTHORIZATION-1.

Topic: **X WINDOW SYSTEM**
SubTopic:

Objective 68

Ensure the setuid and setgid privilege bits are not set on the xterm program.

Rationale:

X is a popular network-based window system that allows many programs to share a single graphical display. The X Window System is a major security hazard. Although there are a number of mechanisms inside X to give some security features, these can be circumvented in many circumstances (Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

As root, execute the following command:

```
/bin/find / -name xterm -exec ls -ldg {} \;
```

Expected Results:

The xterm program is not SUID or SGID.

Comments:

On DII COE perform the same command substituting dtterm for xterm.

Topic: **X WINDOW SYSTEM**
SubTopic:

Objective 179

Verify the systems listed in xhost are appropriate. Determine what release of X is used on the system.

Rationale:

X uses a system called xhost to provide a minimal amount of security for window system users. Each X Window Server has a built-in list of hosts from which it will accept connections; connections from all other hosts are refused. The design of the X Window System allows any client that successfully connects to the X Window Server to exercise complete control over the display. If a person can log into a system, they can capture another user's keystrokes no matter how the xhosts is set (Garfinkel and Spafford, 1992).

Release 4 of the X Window Protocol has a secure feature on the xterm command that makes the window change its color if it is not receiving its input directly from the keyboard. This is a partial fix, but it is not complete (Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

Type the following command to produce a list of which hosts are listed in xhost:

```
%  xhost
```

Expected Results:

Only trusted hosts should be in list returned or the message "Access control enabled, only authorized clients can connect." will be returned.

Comments:

It is preferable that the xhost security not be used and that xauth or another security mechanism be used.

Topic: X WINDOW SYSTEM
SubTopic: Denial of Service

Objective 182

Determine if the X server is vulnerable to the specified denial of service attack.

Rationale:

Even if the xhost facility is used, the X Window System may be vulnerable to attack from computers not in the xhost list. The X11R3 Window Server reads a small packet from the client before it determines whether or not the client is in the xhost list. If a client connects to the X Server but does not transmit this initial packet, the X Server halts all operation until it times out in 30 seconds (Garfinkel and Spafford, 1992).

DII COE SRS Requirement:

Test Actions:

Step: 1

Required Action:

From a networked host, type the following command:

```
%  telnet <localhost> 6000
%  telnet <localhost> 6001
```

Expected Results:

Should get a message "Unable to connect".

If the X server has a problem, the workstation's display will freeze. In some X implementations, the X server will time out after 30 seconds and resume normal operations. Under other X implementations, the server will remain blocked until the connection is aborted.

Comments:

The denial of service vulnerability should not exist.